

Mapua Institute of Technology
Intramuros, Manila

**Indoor Biometric Security with SMS alert
System and Electronic Logbook**

by

**John Michael J. Bernabe
Jessica Mae S. Salgado
Mirriam Joy C. Sorreda**

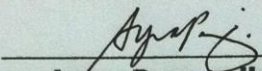
A Design Report Submitted to the School of Electrical Engineering, Electronics
Engineering, and Computer Engineering in Partial Fulfillment of the Requirements
for the Degree

Bachelor of Science in Computer Engineering

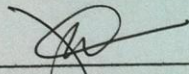
Mapua Institute of Technology
March 2012

Approval Sheet
Mapua Institute of Technology
School of EECE

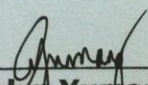
This is to certify that I have supervised the preparation of and read the design report prepared by **John Michael J. Bernabe, Jessica Mae S. Salgado, and Mirriam Joy C. Sorreda** entitled **INDOOR BIOMETRIC SECURITY AND SMS ALERT SYSTEM WITH ELECTRONIC LOGBOOK** and that the said report has been submitted for final examination by the Oral Examination Committee.


Ayra Panganiban
Design Adviser

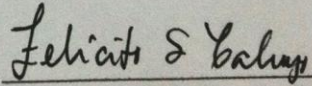
As members of the Oral Examination Committee, we certify that we have examined this design report, presented before the committee on **March 7, 2012**, and hereby recommended that it be accepted in fulfillment of the design requirements for the degree in **Bachelor of Science in Computer Engineering**.


Joshua Cuesta
Panel


Carlos Hortinela IV
Panel


Analyn Yumang
Panel

This design report is hereby approved and accepted by the School of Electrical Engineering, Electronics Engineering, and Computer Engineering in partial fulfillment of the requirements for the degree in **Bachelor of Science in Computer Engineering**.


Felicito S. Caluyo
Dean, School of EECE

Acknowledgement

First, the group would like to thank the Almighty Father for His guidance, undying blessings, and for giving us the strength, perseverance, good health, and wisdom throughout this course.

To acknowledge our dear families for their love, support, understanding, prayers, and encouragements while doing this project.

To our instructor, Engr. Lilibeth Mendoza and adviser, Engr. Ayra Panganiban, for helping us, giving us pieces of advice, and being patient during consultation hours.

To our friends, who sincerely prayed for the success of the project, and indeed a big help to all of us.

John Michael Bernabe

Jessica Mae Salgado

Mirriam Joy Sorreda

Table of Contents

Title Page	i
Approval Sheet	ii
Acknowledgement	iii
Table of Contents	iv
List of Tables	vii
List of Figures	viii
Abstract	xi
Chapter 1: Design Background and Introduction	1
Background	1
Customer	3
Needs	3
Solution	5
Impact	6
Constraints	7
Differentiation	8
Benefits	10
Definition of Terms	11
Chapter 2: Review of Related Design Literatures and Studies	13
Chapter 3: Design Procedure	17
Hardware Development	19

	Block Diagram	20
	Digital Persona	21
	Circuit Diagram	24
	Software Development	25
	Use Case	25
	Database Schema	39
	Activity Diagram	40
	Prototype Development	45
Chapter 4:	Testing, Presentation and Interpretation of Data	47
	System Testing	47
	Failure to Enroll Rate	47
	Reading Test	49
	Failure to Capture Rate with Conditions	51
	Log In /Log Out Accuracy Test	53
	Error Occurrences and Encounters	54
	Program Testing	55
	SMS Gateway	55
	Accounts Management	56
	Run Monitoring per Door	58
	Daily Time Records Log	60
	Change Password	61
	Generate Reports	62
Chapter 5:	Conclusion and Recommendation	63

Conclusion	63
Recommendation	64
References	64
Appendix:	67
Appendix A: Operation's Manual	67
Appendix B: Pictures of Prototype	80
Appendix C: Program Listing	82

List of Tables

Table 4.1 Failure to Enroll Rate	48
Table 4.2 Failure to Capture Rate	50
Table 4.3 Failure to Capture Rate with Conditions	51
Table 4.4 Log In/ Log Out Accuracy Test	53
Table 4.5 Expected Events and Output for the SMS Gateway	55
Table 4.6 Expected Events and Output for the Account Management	56
Table 4.7 Expected Events and Output for the Run Monitoring	58
Table 4.8 Expected Events and Output for the DTR Logs	60
Table 4.9 Expected Events and Output for the Change Password	61
Table 4.10 Expected Events and Output for the Generate Reports	62

List of Figures

Figure 3.1 Design Procedure	18
Figure 3.2 Block Diagram	20
Figure 3.3 Circuit Diagram of the Relay Driver	24
Figure 3.4 Use case name: Run Monitoring	25
Figure 3.5 Use case name: SMS Gateway	27
Figure 3.6 Use Case Name: View Individual Logs	28
Figure 3.7 Use Case Name: Printing of Individual DTR	29
Figure 3.8 Daily Time Record Logs	33
Figure 3.9 Change Password	34
Figure 3.10 Display Reports	35
Figure 3.11 Administration Log in Form	37
Figure 3.12 Database Diagram	39
Figure 3.13 Logs In/Out	40
Figure 3.14 Admin Users Main Form	40
Figure 3.15 Owner User Main Form	41
Figure 3.16 Main Form of the HR user	41
Figure 3.17 SMS Gateway	42
Figure 3.18 Account Management	42
Figure 3.19 Run Monitoring	43
Figure 3.20 Daily Time Record Logs	43

Figure 3.21 Change Password	44
Figure 3.22 Generate Report	44
Figure A.1 Account Login	68
Figure A.2 Account Login Successful	68
Figure A.3 Admin Main Interface	69
Figure A.4 Configuration of Modem Settings	69
Figure A.5 Device Manager	70
Figure A.6 Modem Connection Successful	70
Figure A.7 Modem Connection Error	70
Figure A.8 Accounts Management Interface	71
Figure A.9 User Registration Form	71
Figure A.10 Fingerprint Template Registration	72
Figure A.11 Fingerprint Template Registration Successful	72
Figure A.12 Record Selection Error	73
Figure A.13 Update Successful	73
Figure A.14 Delete Verification	73
Figure A.15 Delete Successful	74
Figure A.16 DTR Form	74
Figure A.17 Log In Successful	75
Figure A.18 Log Out Successful	75
Figure A.19 Permission Denied	75
Figure A.20 Change Password	76

Figure A.21 Change Password Successful	76
Figure A.22 Change Password Error	76
Figure A.23 DTR Logs	77
Figure A.24 Report Generation	77
Figure A.25 Report Generation	78
Figure A.26 Modem Not Connected	78
Figure B.1 Power Supply, Relay Driver and Parallel Port	80
Figure B.2 Mini-Door	80
Figure B.3 Fingerprint Scanner	81
Figure B.4 Modem/Broadband	81

Abstract

The design is a security system that can record data from all people who enter and leave the building which has an office and a household. The system is made to impose tighter security within the premises. The user will scan their fingerprint on the biometric device to enter a specific floor area. The system has levels of access which will restrict the users to enter places wherein they don't have the privilege. The system consists of a biometric fingerprint scanner, electric door lock, and an SMS module. The software will be responsible for the commands sent to the door lock when a fingerprint is scanned. This software is installed on a desktop computer running on a windows operating system. The attendance monitoring is also covered by the system. A broadband is used for the SMS notification of the system.

Keywords: Biometric Fingerprint Scanner, Electric door lock, Software, SMS Module, Windows Operating System,

Chapter 1

DESIGN BACKGROUND AND INTRODUCTION

The introduction gives an overview of the design project, giving the reader the background of the problem to be reported.

The indoor fingerprint security with SMS alert and electronic logbook comprises of a hardware where a biometric fingerprint sensor is used, and software which the user will use for controlling the hardware.

Background

Safety is the condition or state of having the freedom from failure, damage, error, accidents, harm, injury, loss or any event considered to be undesirable. In order to be safe, one has to be secured. Security is the form of protection against any undesirable events. According to the Institute for Security and Open Methodologies (2008), security is a form of protection where a separation is created between the assets and the threat. IT, physical, political, and monetary are some of the types of security. Physical security includes home, school, food, and infrastructure securities. "Prevention is better than cure" this famous quotation can also be applied in security. In order to prevent any risks of danger, harm, or loss one must have security. Some examples or forms of securities include door locks, security guards, anti-virus applications, and passwords.

In terms of security, nowadays biometric authentication is considered as more reliable compared to the traditional security such as password based and lock-and-key. Biometric comes from the Greek words "bios" and "metron" which means life and measurement. It refers to the technology of identifying and/or authenticating a person using distinct human body characteristics such as face, fingerprint, DNA, palm print, iris, retina, and voice. This biological identification technology provide higher sense of security compared to the traditional ones for the reason that these unique marks or features in the body cannot be given to someone else to use and these cannot be lost or misplaced for it is something that people have all the time. To consider a biometric trait as a reliable and secured option for authentication, it should possess universality, distinctiveness, permanence, and collectability. Other criteria for assessment are performance, acceptability, and circumvention. There are two types of biometric characteristics - physiological and behavioral. Physiological identifiers refer to the physical biometric traits of an individual. Physiological biometrics include DNA, fingerprint, facial recognition, hand geometry, ear recognition, iris, and retinal scan. On the other hand keystroke, signature, and voice recognition are behavioral biometrics. This type of biometric measures the traits acquired naturally by an individual over time.

Among all the biometric traits, fingerprint is one of the oldest methods used for various practices. Back in the 14th century, China used fingerprints to distinguish one individual to another. Fingerprint identification is the most widely used of all the biometric devices because of its uniqueness and consistency over time. Applications of biometrics include computer login, access to office buildings and homes, protect personal property, etc.

Customer

A customer, Mr. Jhayson Allen S. Tan, the son of one of the stockholders of Concord Metal Inc. whose family lives in the same building where there is the office of the company is located. The company is a machine shop and handles metal works business. The compound is located in Caloocan City. It is a six-storey building composed of an office and a residence area. There are three different families, who are living in the compound. Each family owns a house in different floors. There are two ways to access the floors: the first one is through the use of the elevator, and the second one is through the stairs. In order to enter each floor there is a door that one needs to enter to fully access the facility of the floor.

Needs

The customer needs a solution regarding their problem about robbery. A few months ago, an anonymous group of people infiltrated their compound. It was said that the thieves used the door keys to open the locked doors and waited for the right time to steal valuable things. To resolve the robbery case, the residents of the building tried changing the keys to the entry points of the whole building. Unfortunately, it was ineffective because another incident of stealing occurred. Mr. Tan, the customer also tried installing CCTV cameras in their building. He installed cameras in the garage, office, elevator entrance, and to the main gate, unfortunately another incident of stealing occurred. He found out that the thieves could access the building by climbing the house next to his and not in the main gate. The thieves could also access different kinds of floor and just waited for the right chance to steal some valuable things if no one was around. In this context, the customer needed to strengthen the security for his house and properties in order to prevent thieves from accessing his compound.

Given this situation, the other families wanted a security system that would monitor the people from coming in and out of their compound. They also wanted a system if could inform them that an unauthorized person would attempt to enter the compound. Mr. Tan also wanted to have the assurance that only authorized persons could have access or by any would enter the compound.

Every biometric system has the hardware and software combinations. This kind of system is not fully functional if the software component of the system is not present, hence hardware and software go hand-in-hand. The software is the factor responsible for transforming the data acquired in the hardware component into useful information. This information will be stored in the database for future use or reference. Databases are very useful in terms of storing large amounts of data. Through the integration of database into the system, querying of data will be faster, thus saving more time compared to the manual acquisition of data.

To achieve a better security system, additional features are incorporated into the existing system. One of the add-ons is the alert system. Alert systems are used to relay a message, warning, or notification. An alert system will help the user be informed of the current situation. The alert systems are intended to notify or alarm the user so that he/she can do the appropriate action/s right away.

Solution

Having the need to impose better and tighter security measures, the group aim to design an indoor fingerprint system that will address the shortcomings of the old method of security. The following are the specific objectives of the project:

- To build a security system using a fingerprint scanner connected to a PC that will only allow authorized residents/people to enter all areas and openings of main doors.
- To design and to make a software application that will limit a person's access based on his/her authorization.
- To log the daily time record of the people coming in and out of the compound in a database.
- To interface an SMS module to the system for notification purposes.

Impact

This design will have a strong impact on the safety of the residents in the area as well as meet their needs for protection through the access privilege applied. They need not worry if they forget to lock their door or not because the design is programmed to lock the door automatically, and it can only be opened by the person who has access to it.

On the other hand, this design will be helpful to the company economically as the system will not involve the paper and pen method, the system will save all the details of the person who goes in and out of the compound, and will also help inform them of any unauthorized or unknown person trying or attempting to enter the compound. With regard to social aspect,

the residents will feel more secured since a tighter security system will be using modern technology and not only relying on the mechanical system.

Constraints

The constraints of the design are being wired and PC based. All entry points in the compound can only be accessed by the people, who are registered in the database. They can use only one finger print either the left hand or the right hand to give information to the database. All visitors can have access as long as they are accompanied by an authorized person. However, the design will not directly give the necessary "security" that they need but this will help them to be informed if someone is trying to infiltrate their compound. In addition to that, the instance of robbery from their place will be averted.

Using the fingerprint biometric scanner, concerns about cleanliness will also arise. By placing the finger on the scanner, germs will be passed from one person to another. Due to this condition, it is advisable to occasionally clean the biometric fingerprint scanner.

The biometric scanner only has its testing program, and because of this the group decided to make their own program for the biometric to work. The language that will be used for the programming of the biometric is the Visual Basic .Net. The program will include the adding, deleting, editing, and searching

of users. In addition, the SQL will also be used for the database. The database will store the record of the log in and log out of those in the compound. The program will also store whether a person who logged-in at a certain floor successfully entered a room or not. The data stored in the database will also include information of those under the company premises, such as their time-in, time-out, number of times logged-in and out, and as to what floor they entered.

The notification of any unauthorized person attempting to enter the premises is only done through sending text messages. Hence, only those who are subscribed to the SMS alert system will know if someone is trying to enter an area where the person has no access.

The designers used an electronic door lock with manual override which means in case of emergencies, such as brownout, the door can still be opened using a key. However, this can also be the weak point of the door lock as intruders can steal and/or forge or duplicate the keys or pick the door lock to be able to enter the building or area without using the biometric scanner.

Also, the system does not have the ability to check whether the fingerprint is already existing and registered in the database.

Differentiation

Bayometric introduces BayLock Outdoor Biometric Fingerprint Access Control System, the world's smallest IP-based fingerprint terminal with time attendance and access control capability. From a simple door control to complex network environment, BayLock Outdoor Fingerprint Device supports full functionality of time attendance and access control.

The difference of indoor fingerprint security using SMS alert and electronic logbook design from other existing systems is the different privilege of access programmed in the system. The system can give different privileges access to different floors using the type of person which the administrator assigned to a specific person.

There are many kinds of person type who have a specific privilege access in the building. The first person type will be for the guard, who has the privilege only for the main gate. The second person type is the employees, who have the privilege to the main gate and at the second floor. The third type of person is for the residents, who have the privilege to enter the main gate and the third floor. The fourth type of person is for the owner, who has the privilege to enter all the premises of the building. There is also a special case wherein an employee is also a resident of the building. In this case, this type of person also has the ability to enter all the premises.

This design can also record entry information of a person who enters the perimeter. If the person attempts to enter the restricted areas, their credentials will be recorded and be informed to the registered person that an unauthorized person is trying to enter the areas which are out of their given privilege areas. The entry information will still be recorded even though they are not able to enter a specific place. There will be three chances to repeat the Biometric input. If the person exceeds the chance the system has given, it will inform the registered person/s through an SMS alert.

The summary of the Daily time record of the employees or everyone who are registered in the system will also be generated. There is also a different account for the owner, human resources on time and administrator who has a corresponding right in the system.

Benefits

The design will benefit the residents as well as for business security. The company will be secured and the residents, too because of the access privilege to be implemented. In addition, those people who are not registered in the system can no longer enter the area without the help of a registered person. This will also be beneficial to them as they will be informed if some unauthorized people are trying to enter a prohibited area. The logbook has two purposes the first, it will record the information of the people coming in and out of the

premises for both the residents and the company; second, it will serve as a record keeper for the employees attendance whether they reported for come to work on time or not, and went home early or came in late.

The residents will greatly benefit from this design as using the biometric fingerprint scanner will serve as a source of identification whether the person is authorized to enter the area, to protect their properties, and to avoid robbery.

Definition of Terms

Biometric – the measurement of physical characteristics, such as fingerprints, DNA, or retinal patterns to in verifying the identity of individuals.

SMS module – the module needs a GSM modem to send or to receive the SMS messages. Usually, these kinds of modem are externals, linked to the machine via serial cable.

Electronic Door Lock – the device that operates using an electric current. Some systems are mounted directly to the lock, and the only security device employed while others are connected to an access control system.

Fingerprint – an impression of the markings on the inner surface of the last joint of the thumb or other finger.

Hardware – the mechanical equipment necessary in conducting an activity, usually distinguished from the theory which the design can make the activity possible.

Printer Port – is also known as the parallel port. It is a type of interface found on the computer for connecting various peripherals. With the bi-directional version of the port, this allows the transmission and reception of data bits at the same time.

Software –the programs used to direct the operation of a computer as well as documentation which give instructions on how to use these.

Chapter 2

Review of Related Design Literatures and Studies

This chapter deals about studies the researchers considered for the development of the Indoor Fingerprint Security with SMS alert and Electronic Logbook. The researchers used the following articles to give the readers an overview of how the design was developed by using some of the data being gathered which helped in the construction of the prototype. Further, the reviewed articles gave the researchers a concise understanding of the advantages and disadvantages that may take place in the design of the hardware.

From the article entitled "A Guide to Biometric Fingerprint Sensors: Major Manufacturers and Technical Specifications" by Michelle Shen, an IT Consultant of ePolyMath.com, a biometric sensor is a fingerprint image capture device that matches the uniqueness of each print read by the sensor and compares it to the one stored in its module or local system database (2002). From this article, the researchers were able to learn the different types of fingerprint sensors and their costs. Among all the sensors, semiconductor sensors are considered to be low cost while optical sensors are considered to have a high degree of stability and reliability, and ultrasound sensors are very precise and fraud-free though expensive to implement.

Moreover, from Michelle Shen's "Vendor Fingerprint Sensors Comparison Chart", the details of the technical specifications of the sensor were discussed as well as those so-called fingerprint application modules which contain fingerprint sensor, middleware and the like (2002). The information in the chart gave the researchers a handy guide to better understand what other developers had achieved, what they were doing, and where they were moving to.

The article "Biometric Embedded Fingerprint Reader Modules" of Kate Hudson discusses a biometric solution to the users of existing equipment with the introduction of the embedded fingerprint reader modules. These are efficient access control system that can be easily integrated into the existing OEM devices to add an improved layer of security (2011).

With this article, the researchers were able to understand these fingerprint reader modules can be used along with the fingerprint software development to provide personalized biometric security solutions and access control systems. These fingerprint readers are equipped with USB, serial or Ethernet interface, and an internal data storage capacity allowing for biometric authentication on the system itself which acts as cutting-edge access control systems. This coincides with the design of the researchers to eliminate the need to carry an access card or remember a PIN while providing a multi-tiered access mechanism to ensure access to those authorized personnel only.

The researchers were also given an idea through the article entitled "Biometric Security System" by Anil Kain which states the recognition of a person by his body, then linking the body to an externally established "identity", forms a very powerful tool for identity management. Since fingerprint recognition is the most widely used method of authentication, the researchers used this biometric technology to control accesses from different persons in the company since this is based on the features found in the impressions made by unique images formed by the human's fingertips (2006).

Another article related to the design which was the "Fingerprint sensor with feature authentication" written by David Kinsella (2005), presents information on how a fingerprint sensor works. Through this article, the researchers were able to acquire knowledge about how the device read the fingerprint of a person as well as to see how detection, analysis and authentication work on the fingerprint sensor.

According to Santhanu Surendran (2007), author of "Biometric Fingerprint Identification", stated by using biometrics, the physiological characteristics of a person can be changed into electronic processes that are inexpensive and easy to use. Also, fingerprinting is the best-known biometric-method of identification that was used for 100 years now. Advances in computer technology and

communication have made even huge fingerprint databases available for instant searchers.

Likewise, since fingerprint-based identification is a method that has been successfully used in numerous applications, the researchers also used it for the design, Indoor Fingerprint Security with SMS alert and Electronic Logbook. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows of a single person. Finger-scan technology is the leading biometric authentication technology in use today with the greatest variety of fingerprint devices available. This technology replaces systems which relies on something a person has in his/her possession, such as a key or ID card or something a person knows such as password or privileged information.

Chapter 3

DESIGN PROCEDURES

This chapter gives a detailed procedure for developing the software and hardware of the prototype. The researchers must follow proper procedures in the development of the prototype, thus thorough discussion of hardware development, flowchart, and components used are discussed here.

Design Procedure

For the design to be developed, basic processes and tools should be taken step-by-step. Figure 3.1 shows a systematic procedure in developing the design topic. First, the group specified what the design topic should be and enumerated the objectives of the design. The group then gathered data and requirements of the design, established scope and delimitations, and collected related information regarding the design topic to visualize the concept as a whole and the prototype to become easier to construct. Based on the researches made, the group identified processes needed to yield the desired outcome of the design. In addition, creation of schematic diagram of the circuit and flowchart were done as well as defining the software features of the prototype.

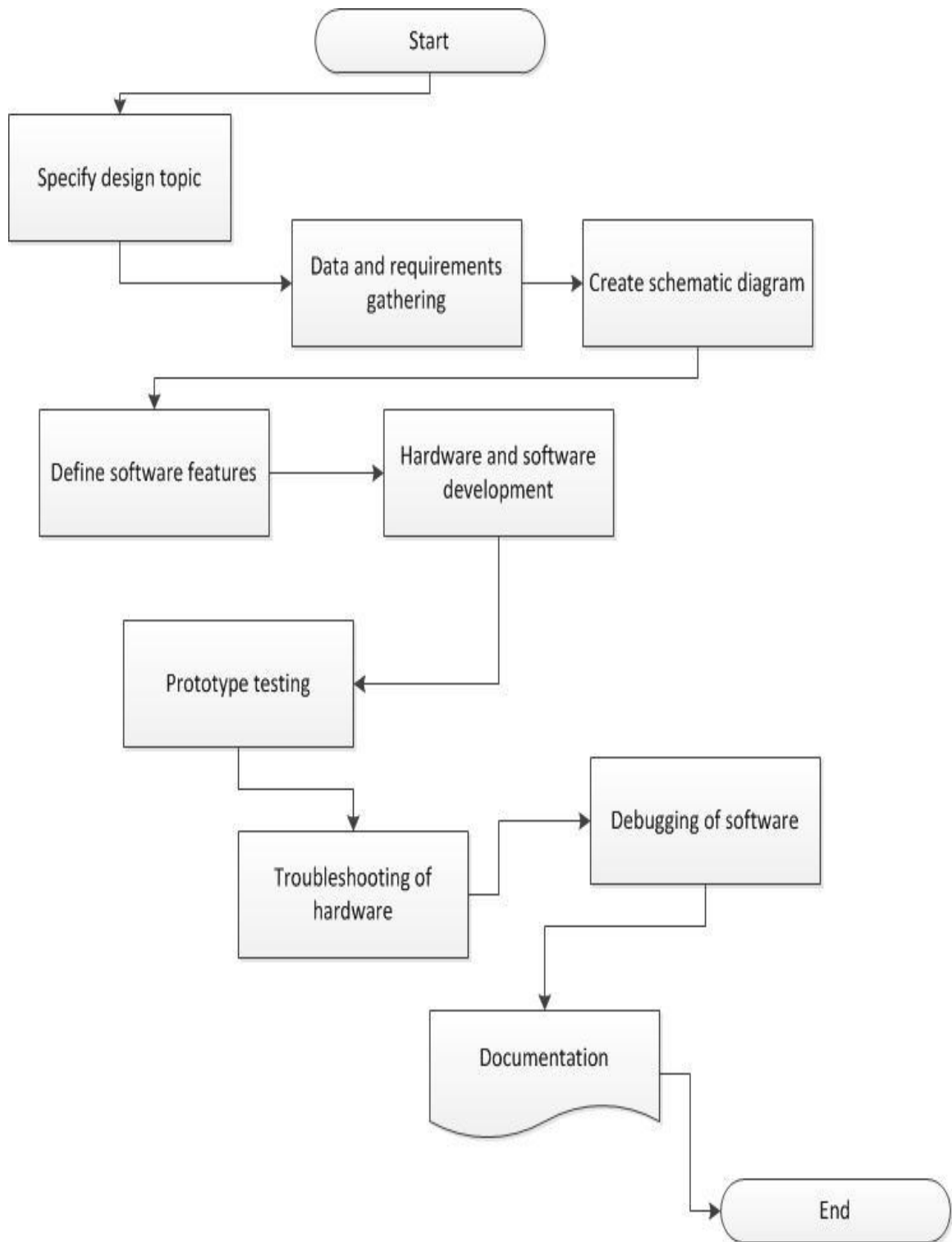


Figure 3.1 Design Procedure

Hardware Development

The group provided the readers the explanation of the block diagram and schematic diagram of the circuit. Figure 3.2 is the block diagram of the prototype that shows the flow as well as relationships of the main components of the device.

The main system worked with Visual basic application, it also had a database linked using SQL. The system functioned with a computer desktop wherein the three biometrics were connected. An SMS module was also installed in the computer desktop for notification purposes. With the use of a parallel data cable, three relay drivers were connected, also supplied with a 220V power source. The relay drivers were then connected to the electronic door lock to supply the required amount of current needed to trigger the door lock to open. A circuit breaker was used to protect electrical circuit from damage due to short circuit or overloading.

The required components were also listed to provide readers a clear visualization and purpose of materials used in designing the prototype. Downloading and studying of data sheets were also done to fully understand the flow of the design which could be implemented properly to the circuitry.

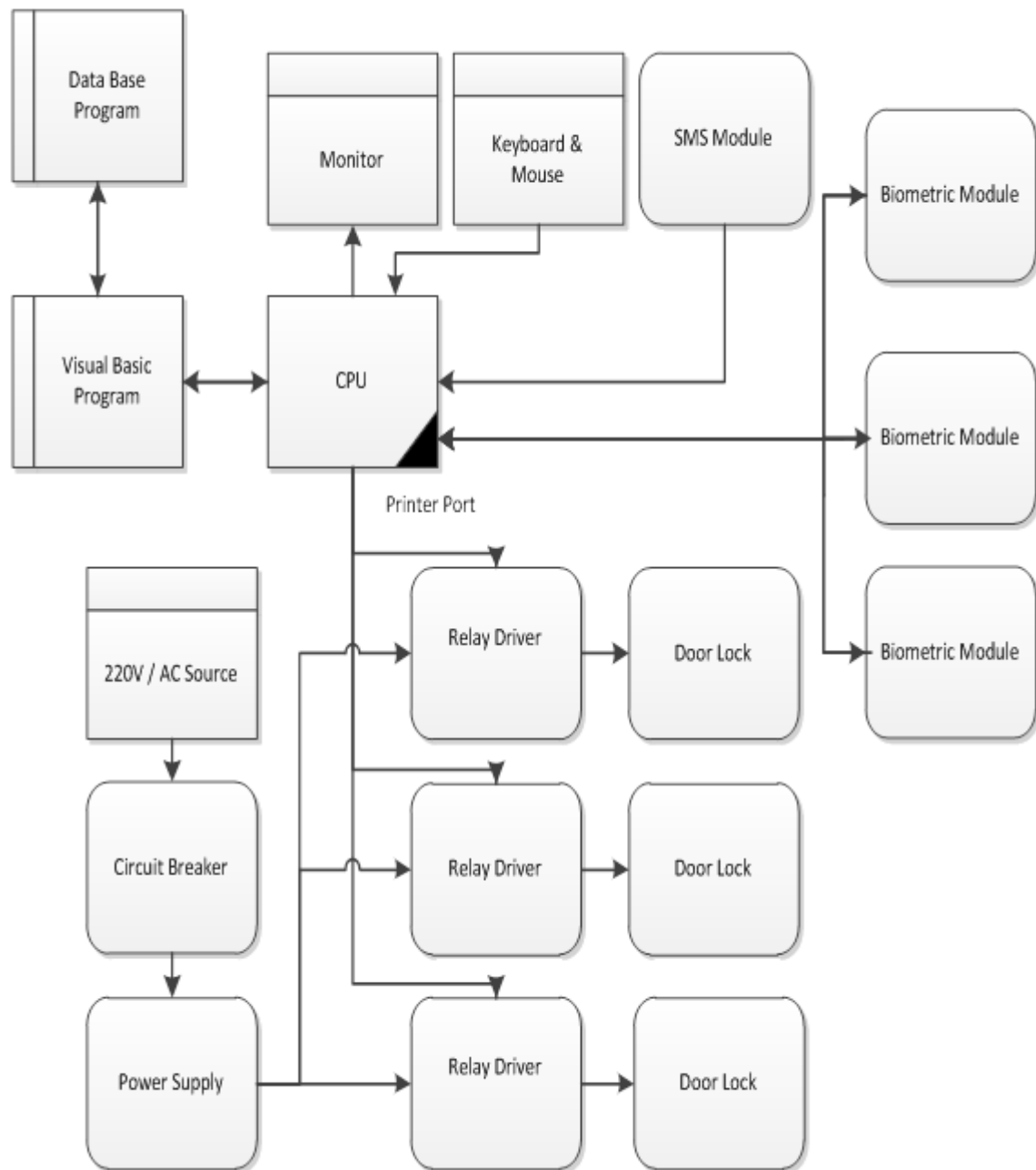


Figure 3.2 Block Diagram

The components illustrated in the block diagram are:

Biometric Fingerprint Scanner – a device that scans and collects the pattern of ridges and valleys that can be found in fingerprints, then converts to a code.

SMS Module – needs a GSM modem to send/receive the SMS messages.

Usually, this kind of modems are externals, linked to the machine via serial cable.

Electronic Door Lock – is a door lock that operates using an electric current.

Some systems are mounted directly on the lock and are the only security device employed, while others are connected to an access control system.

Relay Driver – is a circuit that gives power to an external device.

Power Supply – the device that supplies the needed power for the components to work.

Circuit Breaker – is an electrical switch used to protect electrical circuit from damage due to short circuit or overloading.

Db-25 Pin Connector – is also known as the parallel port made up of several parallel wires. These wires transmit one bit of data at a time so that collectively, they transmit several bits simultaneously.

DigitalPersona “U.are.U 4500 Fingerprint Reader”

The U.are.U 4500 Reader is a USB fingerprint reader featuring an elegant, sleek design with a soft, cool blue glow of course, the unsurpassed performance DigitalPersona is known for. It is made for power-users and shared environments, the 4500 is the natural choice for those who want and need the very best. Here is a look at just some of its features and benefits:

The U.are.U 4500 fingerprint reader is designed for use with DigitalPersona's full range of software: DigitalPersona Pro for Active Directory and DigitalPersonal SDKs for one's own applications.

Some of the many applications and vertical markets in which the U.are.U readers may be used to include:

- Drug dispensary
- Prescription fulfillment
- Time and Attendance
- Point of Service (Retail and Restaurant)
- Health Club membership access
- Finance and Banking account access
- Law Enforcement
- State and Local Government

Easy-to-use

To use, simply place a finger on the reader window and the reader quickly and automatically captures and encrypts the fingerprint image before sending it to the DigitalPersona IDentity Engine for verification.

DigitalPersona products utilize optical fingerprint scanning technology for superior image quality and product reliability. The combination of an U.are.U

4500 Fingerprint Reader with the DigitalPersona IDentity Engine produces an unmatched ability to recognize even the most difficult fingerprints.

Blue LED	Soft, cool blue glow fits into any environment. Provides a pleasing presence; doesn't compete in low light environments, such as restaurants, or conflicts with an alarm condition colors, such as in healthcare.
Small form factor	Conserves valuable desk space.
Rugged construction	High-quality metal casing weighted to resist unintentional movement.
Special undercoating	Stays where one puts it because of a special undercoating.
Rotation invariant	Touches it from any direction, provides a high quality image and matching performance, perfect for shared environments.
Excellent image quality	High-quality optics ensure best image every time.
Works well with dry, moist, or rough fingerprints	Reliable performance over the widest population of users. Reads even the most difficult fingerprints.

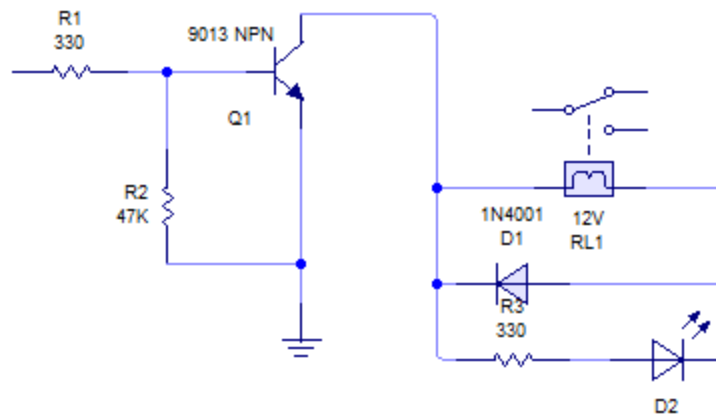


Figure 3.3 Circuit diagram of the relay driver

Figure 3.3 shows the circuit diagram of the relay driver used to design the Indoor Fingerprint Security with SMS alert and Electronic Logbook. The relay driver acts as a switch to send an amount of current needed for the electronic door lock to open. If the current passes through the coil (the one boxed in the diagram), the coil will be energized and will release magnetic force to close the circuit since the path of the circuit is initially open.

Software development

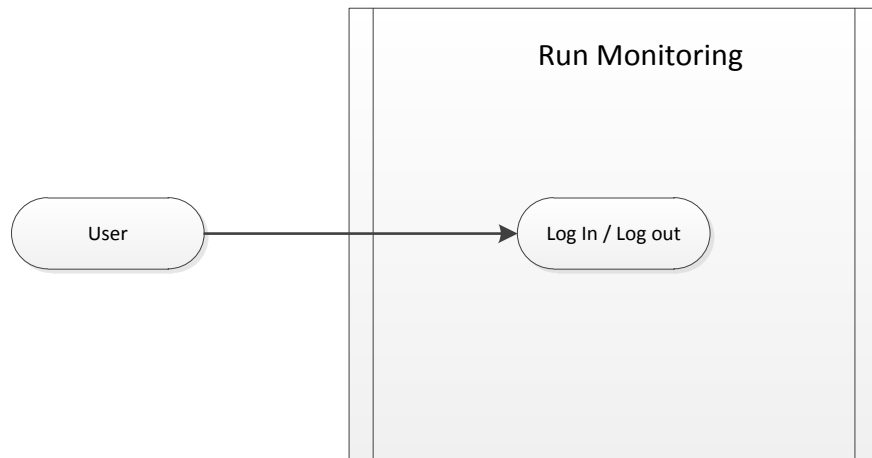


Figure 3.4 Use case name: Run Monitoring

Use case name: Run Monitoring

Pre-condition:

- The monitoring program must already be opened by the administrator
- User's fingerprints enrolled
- SMS module connected
- Door are connected to the printer port
- Door lock has a power source
- Fingerprint reader already initialized

Flow of the program if successful:

- The user must place his / her enrolled finger on the biometric fingerprint scanner
- If the record is found and the access on the said door is granted
- Door lock will open
- Display the following details:
 - a. ID Number
 - b. Name
 - c. Account Type
 - d. Login / Logout
- Exit

Flow of the program if it is not successful:

- If the record is not found
- Count number of attempts
- If attempts reach the limit, send SMS alert
- Exit

Flow of the program if successful but not allowed to enter the area:

- If the user is not allowed to enter the area
- Display the following details:
 - a. ID Number
 - b. Name
 - c. Account Type

d. Inform that the user is not allowed to enter the area

- Send SMS alert
- Exit

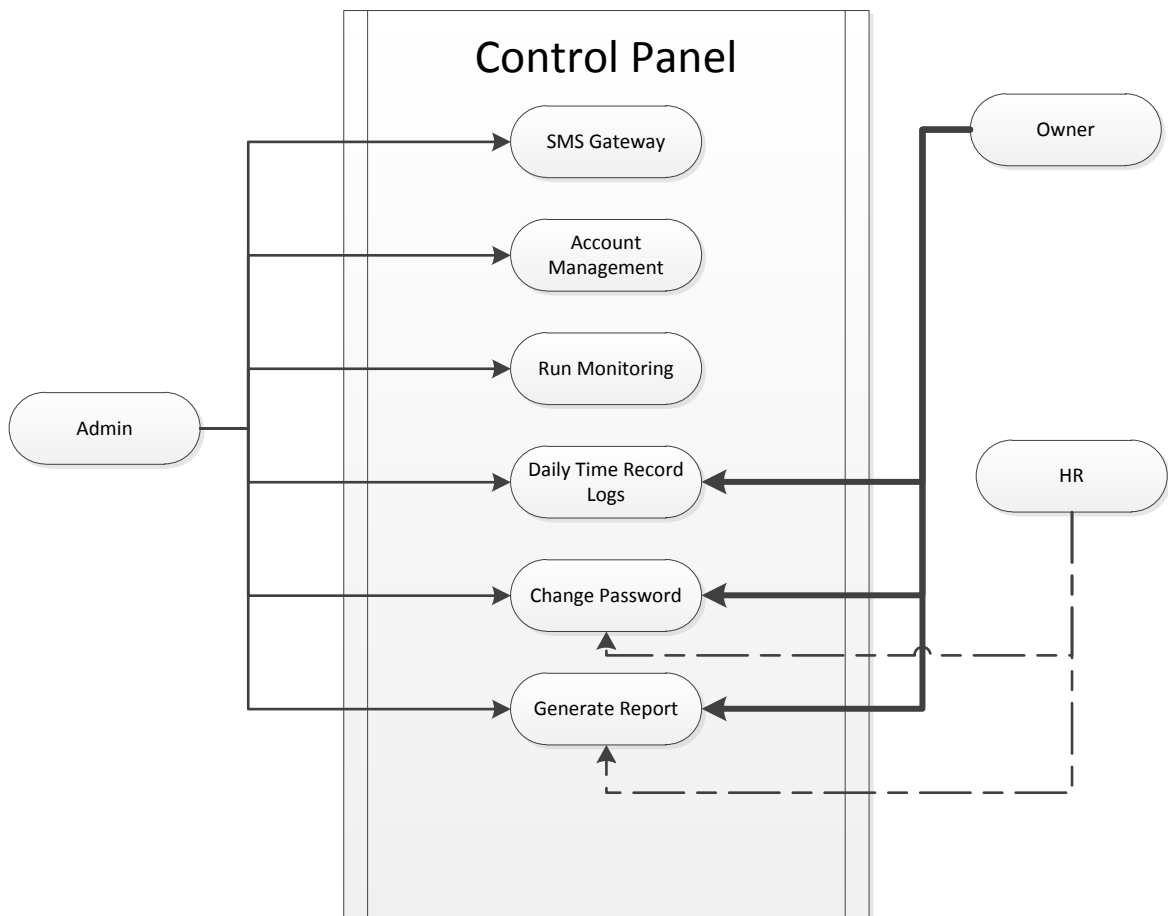


Figure 3.5 Use case name: SMS Gateway

Use case name: SMS Gateway

Pre-condition:

- Connect broadband to the computer
- Close startup of the network provider auto run broadband connection application

- Determine the port being used by the broadband

Flow of the program:

- Select the comm port number used by the broadband
- Select baud rate
- Click connect
- Exit

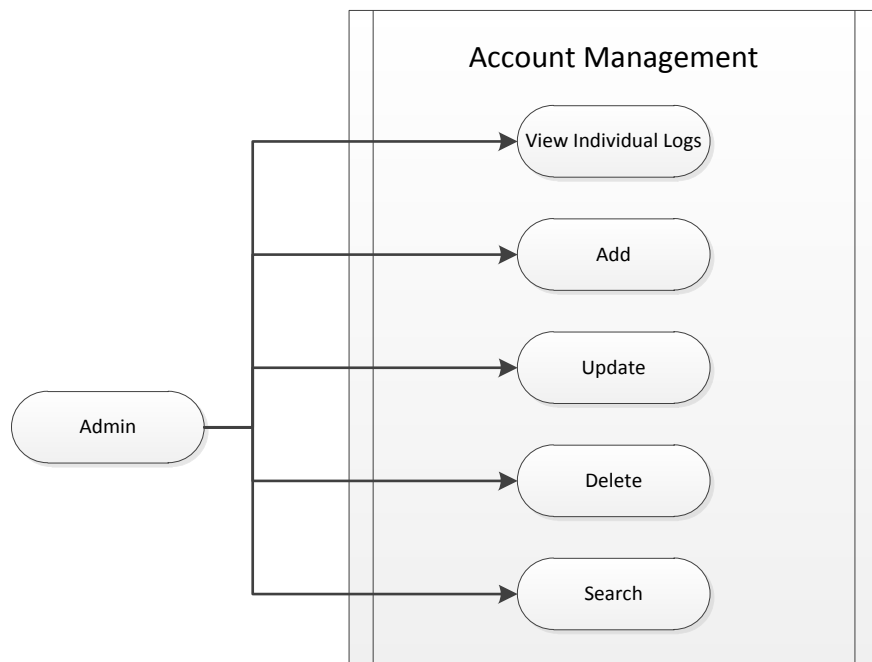


Figure 3.6 Use Case Name: View Individual Logs

Use Case Name: View Individual Logs

Pre-Condition:

- Choose the record the user wants to view
- Click individual logs

Flow of the Program:

- Display the following details
 - a. Detail ID
 - b. Id Number
 - c. Full Name
 - d. Account Type
 - e. Login and Logout
 - f. Door that the user entered
- Exit

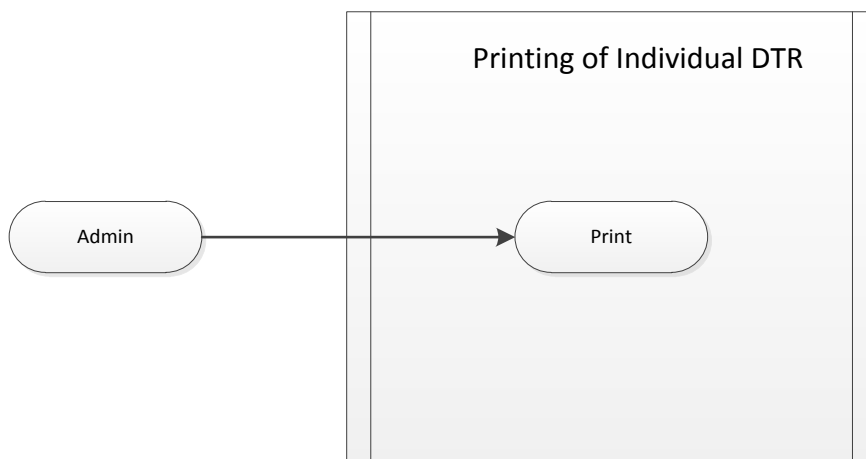


Figure 3.7 Use Case Name: Printing of Individual DTR

Use Case Name: Printing of Individual DTR

Pre-condition:

- There should be a printer installed or connected to the computer

- The printer should be opened
- Printer should have a bond paper and an ink to print the DTR

Flow of the Program:

- Click the Print button
- Enhance Print Preview Dialog will appear
- DTR Ready to Print
- Print the DTR
- Exit

Use Case Name: Add of new user

Pre-condition:

- Finger print scanner should be installed and properly working
- Fill up the form completely and correctly
- Determine if the user will receive SMS alert or not

Flow of the program:

- Fill up the forms completely and correctly
- Click save
- Choose the finger that will be used to register account
- Scan finger for four times to verify
- Registration successful

- Exit

Use case name: Update user

Pre-condition:

- Choose the account that needs to be updated

Flow of the program:

- Choose the user that needs to be updated
- Click update
- Change the necessary information the user wants to update
- Click update
- Exit

Use case name: Delete user

Pre-condition:

- Choose the account that will be deleted

Flow of the program:

- Choose the user that needs to be deleted
- Verify if the selected user will be deleted to the database
- Click ok
- Record deleted
- Exit

Use case name: Search user**Pre-condition:**

- Input the name of the user on the search box

Flow of the program:

- Type the name of the user on the search box
- Display the following details
 - a. Detail ID
 - b. ID Number
 - c. Full Name
 - d. Gender
 - e. Contact Detail
 - f. Birthday
 - g. Username
 - h. Account Type
 - i. SMS Notification
- Exit

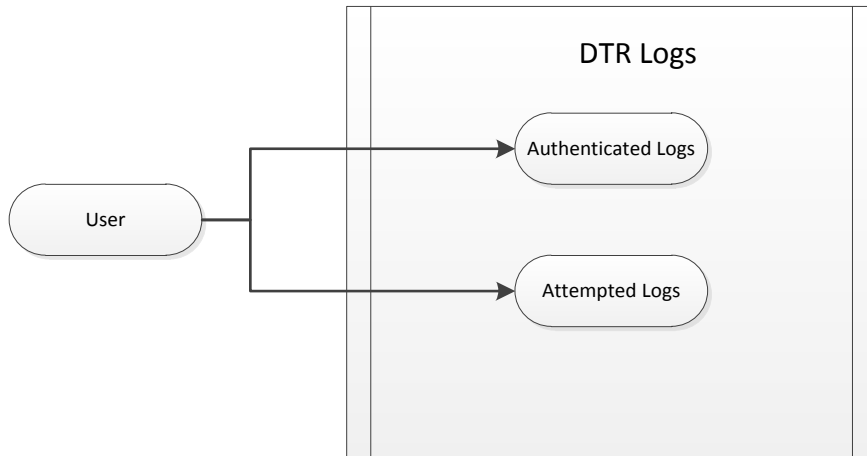


Figure 3.8 Daily Time Record Logs

Use Case Name: Daily Time Record Logs

Pre-Condition:

- Choose the logs to be viewed
- Determine the date the user wants to generate

Flow of the program:

- Choose the type of logs he/she wants to view
- Determine the date to be viewed
- Display logs on the date the user wants to view
- Click clear to delete logs
- Exit

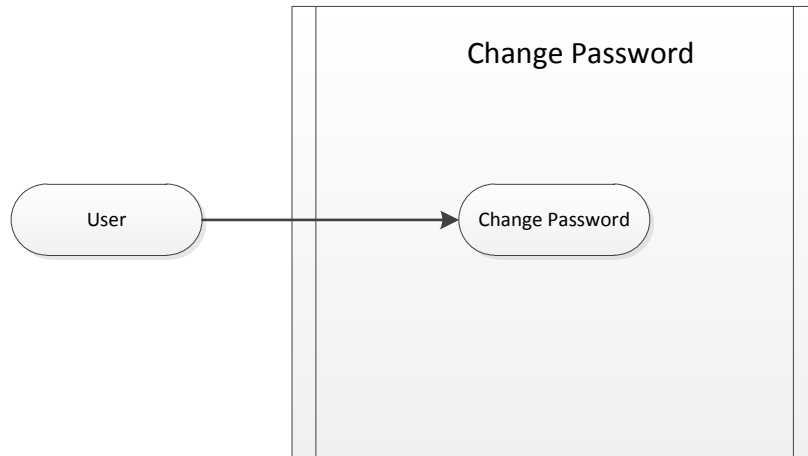


Figure 3.9 Change Password

Use case name: Change password

Pre-condition:

- The user must know his/her old password

Flow of the program:

- Enter the old password
- Enter new password and confirm password
- Press ok
- Save new password
- Exit

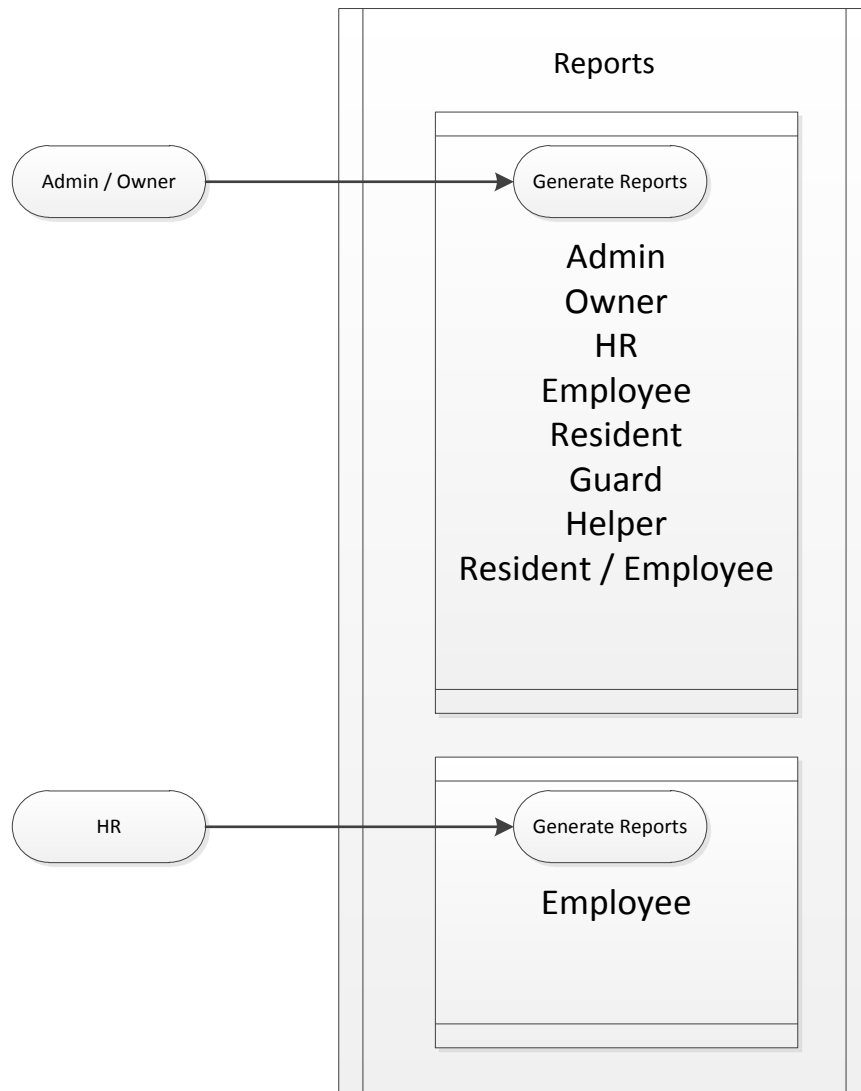


Figure 3.10 Display Reports

Use case name: Display Report

Pre-condition:

- Choose type of person to generate reports
- Administrator and owner can generate all reports
- Human Resource Personnel can only generate employee reports
- Printer must be installed to the computer

- Printer must be opened
- Printer must have a paper and an ink

Flow of the program:

- Select type of person the user wants to display
- Display the following details
 - a. Detail ID
 - b. ID Number
 - c. Name
 - d. Account Type
 - e. Date
 - f. Login and Logout
 - g. Doors that the user used
- Click print if the user wants to print the records
- Print
- Exit

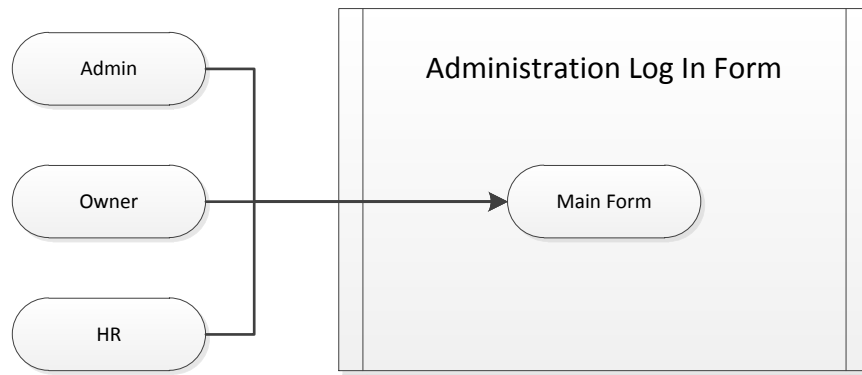


Figure 3.11 Administration Log in Form

Use case name: Administration Log in Form

Pre-condition:

- User must identify what type of user level he/she is assigned

Flow of the program:

- Determine and choose the type of administration he/she is assigned
- Enter username and password
- Click ok
- Main form will be displayed

If administration logged in, this is the list of functions that are enabled

- a. SMS gateway
- b. Account management
- c. Run monitoring
- d. Daily time record logs

- e. Change password
- f. Generate reports
- g. Help
- h. Quit system

If owner logged in, this is the list of functions that is enabled

- a. Daily time record logs
- b. Change password
- c. Generate reports
- d. Help
- e. Quit system

If Human Resource personnel logged in, this is the list of functions that is enabled

- a. Change password
- b. Generate reports
- c. Help
- d. Quit system

Database Schema

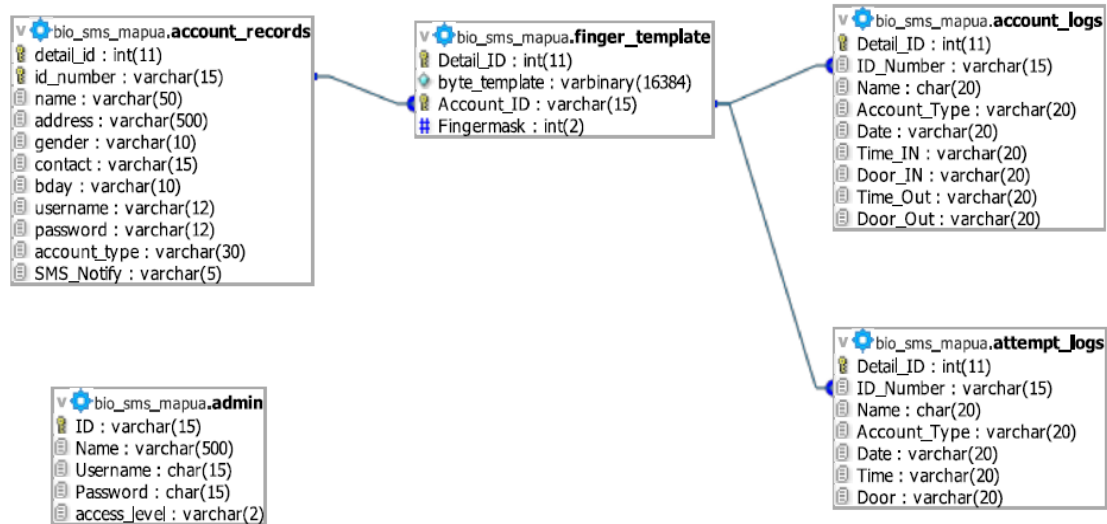


Figure 3.12 Database Diagram

The database diagram in Figure 3.12 shows the database implementation on the desktop application of the system. It consists of five tables which are the account_records, finger_template, account_logs, attempt_logs and admin. One admin account must be added from the database to use the desktop application of the Indoor Fingerprint Security with SMS alert and Electronic Logbook. The account_records table are records of registered users from the system and SMS record to process the sending of SMS notification to alert selected user. The finger_template table are records to store data of the finger print scanner created on a single thumbmark of a user which is registered in the system. The account_logs table are logs created when a user goes in and out of a specific door. Then, the attempt_logs table are logs created when a user attempts to enter an area which is not allowed. Admin table are records of a registered

administrator, owner and Human Resource manager in order to access the user interface of the whole system.

Activity Diagram

This shows the graphical representation of the use case or the workflow of the system. The control flow of the system starts from the initial state to the final state.

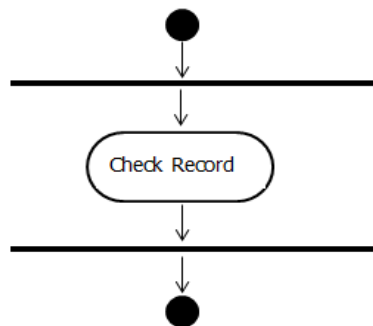


Figure 3.13 Logs In/Out

Figure 3.13 shows the workflow of the log in/out system of the entire user. In the initial state the user will place the registered finger for verification, the fingerprint will be processed by checking the match and access area which the user is allowed.

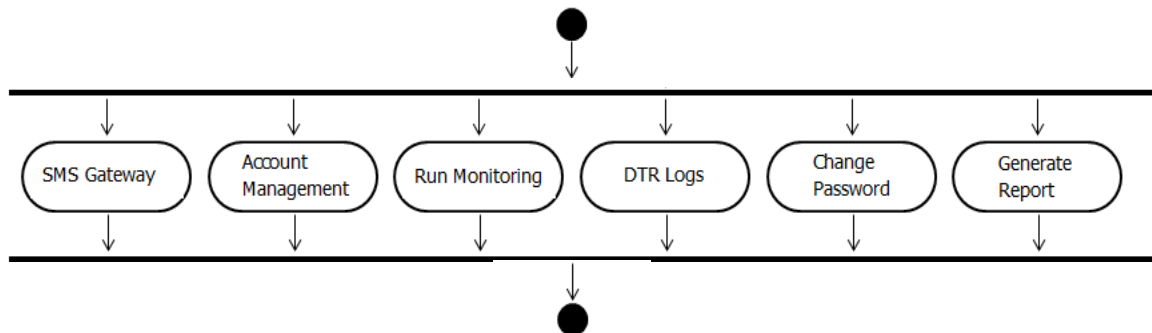


Figure 3.14 Admin Users Main Form

The figure 3.14 above shows the workflow of the main form of the admin user. The user will select from the choices which have different functions.

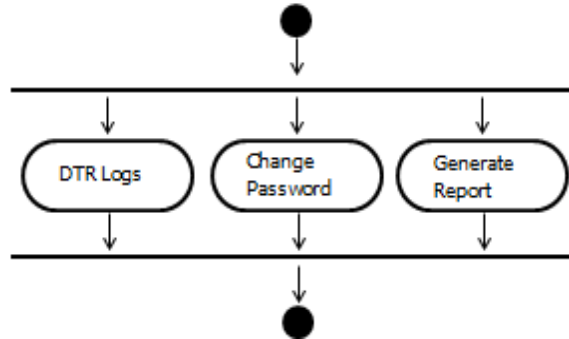


Figure 3.15 Owner User Main Form

Figure 3.15 shows the workflow of the main form of the owner users. This user only has limited functions to work on.

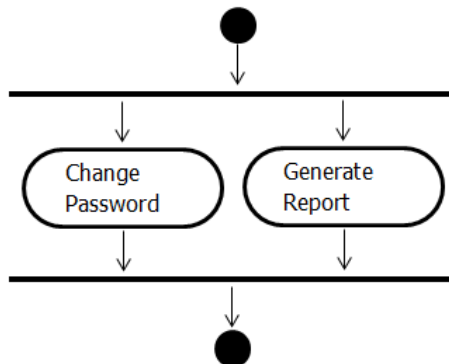


Figure 3.16 Main Form of the HR user

While Figure 3.16 illustrates the workflow of the main form of the HR user. This user is only allowed to change his/her account's password and generates report of the employees' attendance.

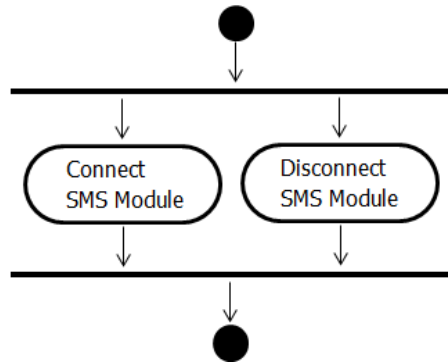


Figure 3.17 SMS Gateway

The figure above shows the workflow of the SMS Gateway for the administrator. The user will select from the choices which port and baud rate needed to connect the SMS module or disconnect the SMS module.

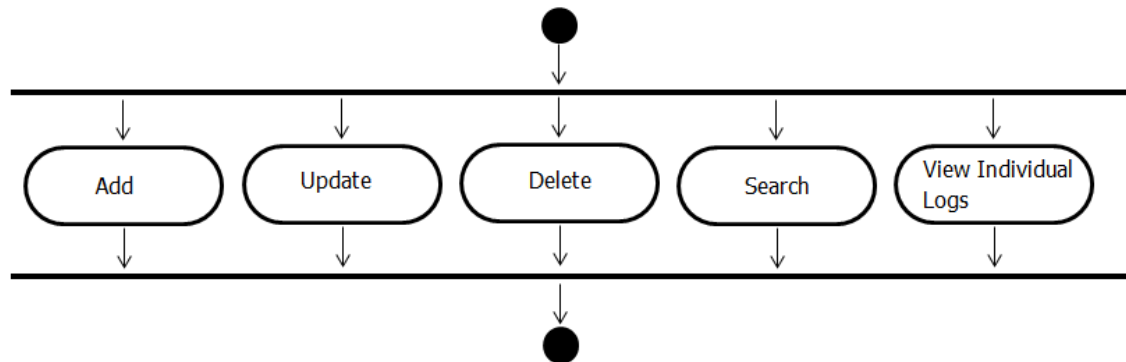


Figure 3.18 Account Management

The activity diagram in Figure 3.18 shows the workflow of the account management of the administrator user. The user will select from the choices whether to add, update, delete, view, search or individual logs.

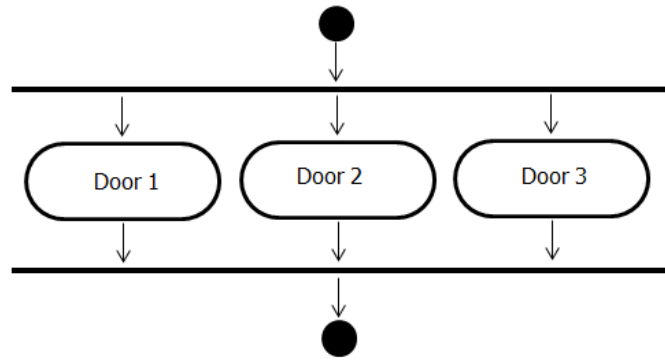


Figure 3.19 Run Monitoring

Figure 3.19 displays the workflow of monitoring the system by the administrator user. The user will select from the doors he/she wants to run.

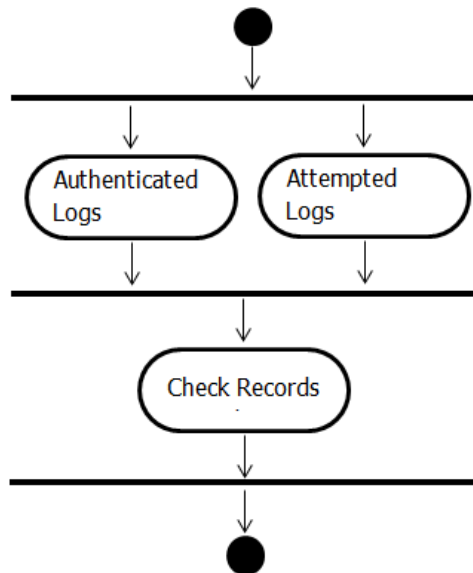


Figure 3.20 Daily Time Record Logs

This figure shows the workflow of the DTR Logs by the administrator. The user will select which logs will be displayed in the form, and he/she will select what date to view, and check the records on the database.

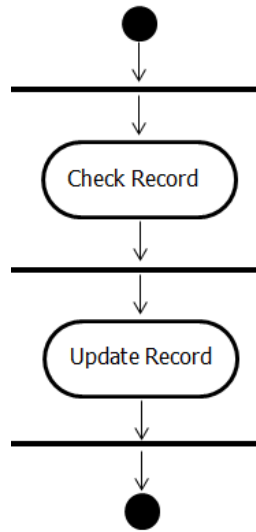


Figure 3.21 Change Password

Figure 3.21 shows the workflow of the change password by the user. The user must input the correct old password recorded from the database to change his/her existing password.

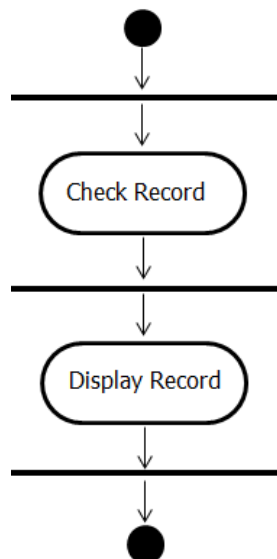


Figure 3.22 Generate Report

The figure above shows the workflow to generate reports. The user will select which category or account type he/she wants to view the generated reports. When verified the user in the database, the record will be displayed.

Prototype Development

The Indoor Fingerprint Security with SMS alert and Electronic Logbook the group designed has the ability to store data of registered users and record the logs of a person when he/she enters or exits the building or a specific area of the building. This can also detect if there is an unauthorized person who is trying to access an area restricted or prohibited to enter. The group will be using biometric fingerprint scanner (hardware) interfaced in three different electronic door locks which will be placed in each entry points of the compound: the main gate, the door office, and the residential door.

For the programming part, the group will be using the language Visual Basic (software) and will link it to SQL (software) for the database. An administrator account will be created for the configuration of the registration of new users. The program can add, edit, update, delete, and remove all users from the system. The administrator can also modify the accounts in the system, change passwords, monitor the logs of all the tenants in the building and generate reports. There will also be an account for HR purposes such as the

attendance monitoring of the employees at the office for evaluation and printed reports. In addition to that, the design will be having an account solely for the OWNER that will have the access to view all the records and can print all date and time records. A user can only have 3 attempts to log-in using the biometric scanner. If limit is reached, there will be an SMS sent to the owner or to the administrator or to a recipient who wished to receive notifications to inform that there is someone who is attempting to enter the building or different floor levels. This attempt will also be logged in the database. A personal computer will be installed at the office located at the second floor of the building.

As regards the levels of access, each person type will be given a specific kind of authorization. All person types, administrator, owner, Human Resource manager, guard, employee, resident, and helper will only be able to access specific areas. The sole person type who has the access on all floors is the owner. While the Human Resource manager and employee can only go through the main gate and the office door. The resident and helper person types will only have access to the main gate and residential door.

Chapter 4

TESTING, PRESENTATION, AND INTERPRETATION OF DATA

This chapter shows the different tests done by the group in line with the objectives enumerated in the first chapter. The testing was done after the development and construction of the prototype to verify if the objectives were accomplished. Furthermore, the following tests exhibited the functionality and reliability of the prototype. The group performed the following tests: compatibility testing of Visual Basic versions and operating systems, project software, and hardware detection test, other software features testing which includes the database of the electronic logbook.

System Testing

Failure to Enroll Rate

The first thing needed for the design to serve its purpose is to store data assigned to the user in which area he / she is allowed to enter. The testing done for registration is called failure to enroll rate. It is the failure of the system to create a proper enrolment template. This is usually caused by low quality inputs. The test is done using the registration of the user by assigning his/her account in different account type to the registration form. The procedures in conducting the test are as follows:

1. set up the entire device in operating condition
2. open the account management and click add

3. fill-up the form, then click the add button to register the fingerprint the user will use to access the assigned area for him/her
4. choose the finger the user will register by clicking one of the fingers on the two hands being displayed on the screen
5. place the chosen finger four times on the fingerprint scanner
6. a message will prompt if the scanning of the finger is successful or failure if the scanning of the finger is a failure, the user needs to scan again his/her finger until the four successful verification of the finger is finished
7. a message will prompt if the registration is successful if the registration is a failure, the scanning of the finger will automatically re-start until the registration is successful.

User	Trial 1	Trial 2	Trial 3
Kim	Success	Success	Success
Ralph	Success	Success	Success
Ian	Success	Success	Success
Sedney	Success	Success	Success
Ken	Success	Success	Success
Hazel	Success	Success	Success
Janica	Success	Success	Success
Kevin	Success	Success	Success

Lorise	Success	Success	Success
Nico	Success	Success	Success

Table 4.1 Failure to Enroll Rate

This table 4.1 shows the registration of a user using the digital persona biometric where the system has high enrollment rate. All registration are successful after placing the users' finger in the scanner four times. The registration on the fingerprint will repeat its process, if the user registration is not successful. This will just finish its process once the registration is successful.

Reading Test

This feature can read the data saved on the program if there is already a registered user. Once there is/are successful registrants, when they tap their finger/s already registered, the program will now read the fingerprint and will welcome the user/s. This compares a person's fingerprint to another one previously recorded in the system database. The computer locates the previous fingerprint by comparing two fingerprints recorded. Once matched from the one recorded previously, the door lock will be activated and opened.

USERS	Kim	Ralph	Ian	Sedney	Ken	Hazel	Janica	Kevin	Lorise	Nico
Kim	S	F	F	F	F	F	F	F	F	F
Ralph	F	S	F	F	F	F	F	F	F	F
Ian	F	F	S	F	F	F	F	F	F	F
Sedney	F	F	F	S	F	F	F	F	F	F
Ken	F	F	F	F	S	F	F	F	F	F
Hazel	F	F	F	F	F	S	F	F	F	F
Janica	F	F	F	F	F	F	S	F	F	F
Kevin	F	F	F	F	F	F	F	S	F	F
Lorise	F	F	F	F	F	F	F	F	S	F
Nico	F	F	F	F	F	F	F	F	F	S

- **S – Success**

- **F - Fail**

Table 4.2 Failure to Capture Rate

Table 4.2 shows the results of the test done using the failure to capture the rate. This performance metric is the probability the system fails to detect a biometric input when presented correctly. The result clearly shows reading a fingerprint using the digital persona biometric has high capture rate of the design. This shows biometric scanner has a 100% capture rate in determining whether a fingerprint is enrolled in a system or not.

Failure to Capture Rate with Conditions

The fingerprint scanner's ability to read the fingerprint will differ from one fingerprint to another. The following tables will show how fingerprint condition will affect the fingerprint scanner's accuracy.

Finger Condition Accuracy Test					
USERS	Correct Alignment of Finger	Incorrect Alignment of Finger	Stained or Dirt on the Finger	Foreign Particle Present on the Finger	Wet Finger
Kim	Success	Success	Success	Success	Fail
Ralph	Success	Success	Success	Success	Fail
Ian	Success	Success	Success	Fail	Fail
Sedney	Success	Success	Success	Success	Fail
Ken	Success	Success	Success	Success	Fail
Hazel	Success	Success	Success	Fail	Fail
Janica	Success	Success	Success	Success	Fail
Kevin	Success	Success	Success	Success	Fail
Lorise	Success	Success	Success	Success	Fail
Nico	Success	Success	Success	Fail	Fail

Table 4.3 Failure to Capture Rate with Conditions

The Table 4.3 shows scanning of a correct alignment of finger has high rate acceptance using the fingerprint scanner of the design. All scanning of fingerprints are successful in all trials from different users.

The second test is the incorrect alignment of finger to be scanned by the fingerprint scanner. Some biometric scanners cannot recognize the user's fingerprint with an incorrect alignment of finger to be scanned. With this test, it shows that the digital persona biometric scanner has a high capture rate that can read and can identify the registered user even if the users place their finger in an incorrect alignment.

On the third test, it shows that the stained/dirt on the finger has a high rate using the fingerprint scanner. This stained/dirt on the thumb is not the problem to have a successful log in or log out using the biometric scanner because there is always a chance that user's finger is dirty or stained in some situations.

The fourth test shows the test with any foreign particle on the thumb using the fingerprint scanner has a good capture rate. In this test the biometric can also read and identify if the user is registered or not, even if there is a foreign particle in the user's thumb or scanner. To avoid this problem make sure the biometric scanner is clean before scanning the finger.

The last acceptance test shows wet finger causes the fingerprint scanner's accuracy very low or bad as the finger is impossible to read. To avoid this kind of problem users must remember the finger must be dried to have a successful log in / log out to the scanner.

The results show all the tests from the fingerprint scanner will also vary from the fingerprint conditions. It also show all tests except wet finger will have a high capture rate of the fingerprint scanner using the digital persona biometric scanner.

Log In / Log Out Accuracy Test

Log In /Log Out Test									
Finger	Trial 1			Trial 2			Trial 3		
Print			SMS			SMS			SMS
#	Scan	Door	Alert	Scan	Door	Alert	Scan	Door	Alert
1	Success	Open	None	Success	Open	None	Success	Open	None
2	Success	Open	None	Success	Open	None	Success	Open	None
3	Success	Open	None	Success	Open	None	Success	Open	None
4	Fail	Close	Sent	Success	Open	None	Success	Open	None
5	Success	Open	None	Success	Open	None	Success	Open	None
6	Fail	Close	Sent	Success	Open	None	Success	Open	None
7	Success	Open	None	Success	Open	None	Success	Open	None
8	Fail	Close	Sent	Success	Open	None	Success	Open	None
9	Success	Open	None	Success	Open	None	Success	Open	None
10	Success	Open	None	Success	Open	None	Success	Open	None

Table 4.4 Log In/ Log Out Accuracy Test

In Table 4.4 the test shows how the door lock, SMS module, and fingerprint scanner interact with each other. The fingerprint is responsible to send signals or data from the program, triggers the door lock or sends SMS alert to the respondent. This table also shows how the door lock and SMS alert response in the result of the fingerprint scanner to the scanning of the thumb of the user.

Error Occurrences and Encounters

This section enumerates the scenarios that might happen and could cause errors to the user or programmer.

- a. Biometric sensor does not emit blue light.
- b. Ports are not configured properly.
- c. .dll file is not pasted to the system file folder of the hard drive.
- d. SMS gateway is not connected.
- e. Printer port is not connected to the computer.

Program Testing

SMS Gateway			
Function	Event	Expected Output	Actual Output
Connect SMS	1. The user open the SMS Gateway	System Launched the SMS Gateway	Ok
	2. Select Com Port		Ok
	3. Select Baud Rate		Ok
	4. Click Connect	Display SMS Device status Connected	Ok
		Close automatically	Ok
Disconnect SMS	1. The user open the SMS Gateway	System Launched the SMS Gateway	Ok
	2. Click Disconnect	Display SMS Device status Disconnected	Ok
		3. Click Close	Close Application Ok

Table 4.5 Expected Events and Output for the SMS Gateway

The table above shows the expected events and outputs for the connection or disconnection of SMS in the SMS Gateway. This shows the expected output for the different steps the user will choose.

Accounts Management			
Function	Event	Expected Output	Actual Output
Search Name	1. The user type the name needs to search	Display Names similar to the user have typed	Ok
	2. The user open the Accounts Management	System Launched the Accounts Management	Ok
View Individual Logs	1. Select the registered user.	User is selected	Ok
	2. Click Individual Logs	Display Log of the selected user	Ok
	3. Click Print	Display Ready to Print PDF File	Ok
	4. Click Close	Close Individual DTR Application	Ok
Add User	1. Click Add	System Launched the Registration Form	Ok
	2. Select Account Type	Enable or Disable ID Number	Ok
	3. Fill up all data's required and click Save	Proceed to Finger print registration	Ok

	4. Incomplete data's required and click Save	Error Message to inform to Fill up all Required Data	Ok
Finger Print Registration	1. Click Save on the Registration Form	System Launched Registration for Fingerprint	Ok
	2. Choose Finger to register	Automatically Directed to Finger Print Verification	Ok
	3. Finger Print Reads finger Input for 4 times	a. Notify User if data was saved	Ok
		b. Display reading if it's not successful	Ok
Update User	1. Select the registered user	User is selected	Ok
	2. Click Update	System Launched the Registration Form	Ok
	3. Fill up all data's required and click Save	Inform user Update Successful	Ok
	3. Incomplete data's required and click Save	Error Message to inform to Fill up all Required Data	Ok

Delete User	1. Select the registered user	User is selected	Ok
	2. Click Delete	Verify user to delete the registered user	Ok
	3. Click Yes	Record Deleted	Ok

Table 4.6 Expected Events and Output for the Account Management

Table 4.6 shows the expected events and outputs of the Account Management to the different functions of the button in the account management form.

Run Monitoring per Door			
Function	Event	Expected Output	Actual Output
Registered User	1. The user click Run Monitoring	System Launched the Run Monitoring	Ok
	2. Users place register Finger to the biometric	Display the user ID	Ok
		Display the Full Name	Ok
		Display the Account Type	Ok
		Display if Log Out or	Ok

		Log In & Save logs	
If Not Registered User	1. Not Registered User Scan Finger to the biometric	Display Not Registered	Ok
		Send SMS Alert	Ok
If User's is Not allowed	1. Registered user attempt to enter unauthorized area	Display the User ID	Ok
		Display the Full Name	Ok
		Display the Account Type & Save logs	Ok
		Display the user is not authorized to enter	Ok
		Send SMS Alert	Ok

Table 4.7 Expected Events and Output for the Run Monitoring

As shown in Table 4.7 the expected events and outputs for the Run Monitoring for the different users, who are registered or not registered in the system.

Daily Time Records Log			
Function	Event	Expected Output	Actual Output
Authenticated Logs	1. The user select DTR Logs	System Launched the DTR Logs	Ok

	2. The user select authenticated logs		Ok
	3. The user select what date to display logs	Display logs on the selected date	Ok
Attempted Logs	1. The user select authenticated logs		Ok
	2. The user select what date to display logs	Display logs on the selected date	Ok
Clear Logs	1. The user click the clear logs		Ok
	2. Verify if the user wants to delete logs	Display Verification Message	Ok
	3. Click Yes	Logs Deleted	Ok

Table 4.8 Expected Events and Output for the DTR Logs

This Table 4.8 shows the expected events and output for the Daily Time Records Log for the different functions in the DTR Log.

Change Password			
Function	Event	Expected Output	Actual Output
Change Password	1. The user select Change Password	System Launched the DTR Logs	Ok

	2. The user enters the old password		Ok
	2 The user enters incorrect old password	Inform User old password is not correct	Ok
	3. The user enters the new password		Ok
	4. The user verify the new password correctly	Inform User new password successfully updated	Ok
	4. The user verify the new password incorrectly	Inform User new password does not match	Ok

Table 4.9 Expected Events and Output for the Change Password

As shown above, the table how change password work and display output in a given situation.

Generate Reports			
Module	Event	Expected Output	Actual Output
Generate Reports	1. The user click Generate Reports	System Launched the Accounts Management	Ok
	2. The user Select Category	Display All Logs of the	Ok

	of the person who wants to view	Category of the person	
		Display detail id	Ok
		Display Id number	Ok
		Display Name	Ok
		Display Account Type	Ok
		Display Date	Ok
		Display Log in/out time	Ok
		Display Door the user entered	Ok
	3. Click Print	Display Ready to Print PDF File	Ok
	4. Click Close	Close Generate Report Application	Ok

Table 4.10 Expected Events and Output for the Generate Reports

This shows how generation of reports works.

Chapter 5

CONCLUSIONS AND RECOMMENDATIONS

This chapter states the overall conclusion of the design which addresses if the objectives were met.

Conclusion

In the design Indoor Fingerprint Security with SMS alert and Electronic Logbook, the designers were able to build a system to address the needs of the customer, Mr. Jhayson Allen S. Tan. The group was able to design a system that would give a better and tighter security to the occupants in the compound. The system built is a PC interfaced fingerprint scanner with a desktop application and SMS module. With this system, only authorized people will be able to pass through the main gate and specific areas of the compound. Levels of access were imposed in the system where the users would only be able to enter the areas assigned to them. Through these levels of access implemented on the biometric sensor, it would help control the access of people coming in and out of the building. Using the electronic logbook, this would register and record the log in and log out information of the person coming in and out of the building. People who were not registered in the database of the program could not enter the building premises.

Moreover, an SMS alert system was incorporated in the system to notify the chosen respondents if there was/were unauthorized person/s who attempted to enter a specific area of the building

Recommendations

To attend to the delimitations of the project prototype, the researchers recommend the enhancement of the project by making the transmission of data wireless so that it is less prone to tampering and it will definitely eliminate the use of long wires and cables. The project supports only fingerprint scanning, so the enhancement may well be as adding a new feature such as eye scanner. Use a USB converter instead of serial data cables and DB-25 pin data cable since most of the CPU's nowadays have few serial ports and sometimes, no printer ports. For the SMS alert system, the researchers recommend the use of gateway instead of modem. However, an internet connection is needed in using the gateway. Compared to the SMS alert system using the modem, the sending of message through a gateway will be faster and will not affect the performance of the computer.

References

1. Adhami, R., Meenen, P (2001). Fingerprinting for Security. *Potentials*, 20, pg. 33-38
2. Dass, S.C., Jain A.K., Yongfang Zhu (2006). Validating a Biometric Authentication System: Sample Size Requirements. *Pattern Analysis and Machine Intelligence*, 28, pg. 1902-1319
3. Faundez-Zanuy, Marcos (2004). A Door-Opening System Using A Low-Cost Fingerprint Scanner and a PC. IEEE A&E SYSTEMS MAGAZINE.
4. Maltoni, D., Maio, D., Jain, A. K., Prabhakar S. (2009). Handbook of Fingerprint Recognition, Second Edition, Springer.
5. Wertheim, K.E. (2010). Human Factors in Large-Scale Biometric Systems: A Study of the Human Factors Related to Errors in Semiautomatic Fingerprint Biometrics. *Systems Journal*, 4, pg. 138-146
6. Herzog P. (2008, July 17). Home Security Methodology. Retrieved from <http://www.isecom.org/research/hsm.html>
7. Hudson K. (2011, May 30). Biometric Embedded Fingerprint Reader Modules. Retrieved from <http://www.articlesbase.com/security-articles/biometric-embedded-fingerprint-reader-modules-4834241.html>
8. Shen M. (2002, September 16). A Guide to Biometric Fingerprint Sensors: Major Manufacturers and Technical Specifications. Retrieved from <http://www.tmcnet.com/biomag/features/shen0902.htm>

Appendix

Appendix A

Operation's Manual

1. System Requirements

- a. Windows XP or later versions
- b. 220V Source
- c. System Unit with Parallel Port
- d. Broadband

2. Installation Procedures

- a. Connect and Install the Digital Persona Fingerprint going to be used for the set-up.
- b. Plug and Install the Broadband to the Computer Device going to be used for the set-up
 - c. Connect the parallel port to the system unit.

3. User's Manual

After installation of the components, the system is now ready to be used. Open the program to start using the System.

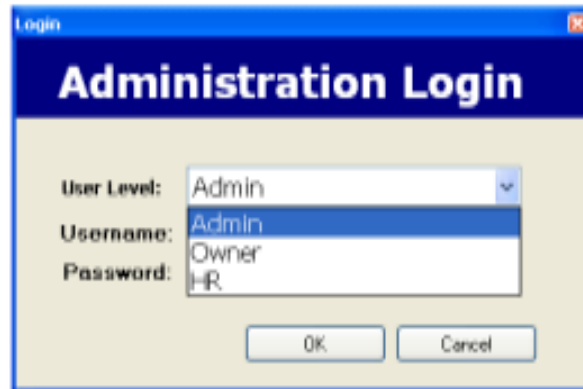


Figure A.1 Account Login

Figure A.1 shows the interface where the admin, owner and HR will login. The user must enter the correct user name and password to be able to continue.

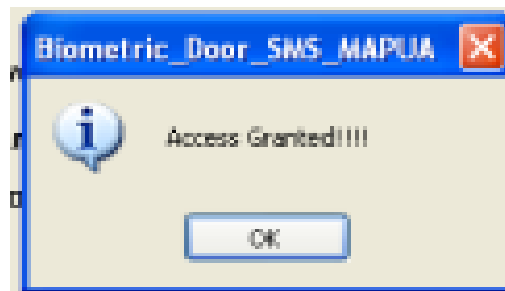


Figure A.2 Account Login Successful

Figure A.2 shows the message displayed when the attempt to login an account is successful.

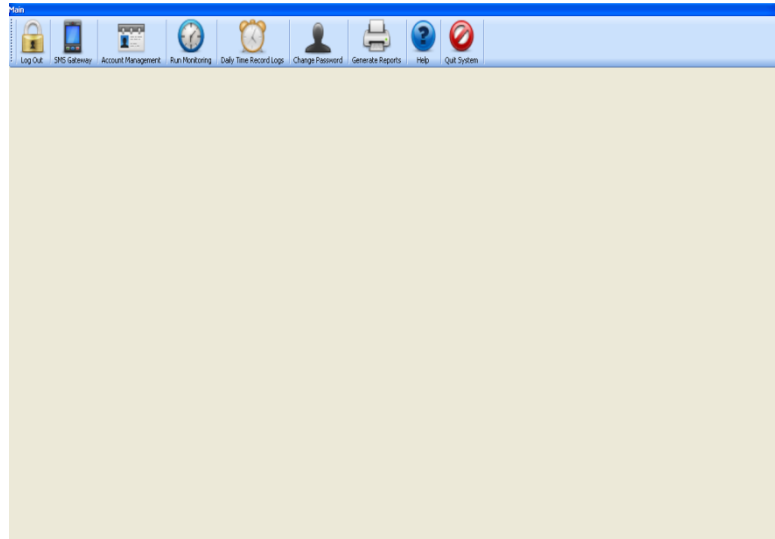


Figure A.3 Admin Main Interface

Figure A.3 displays the main interface once the admin account is logged in. All the primary actions in the system are done by the admin.



Figure A.4 Configuration of Modem Settings

Figure A.4 illustrates the interface when changing the settings of the modem. To identify what comport number to use, go to Device Manager and look for the comport number that the modem uses for its PC interface.

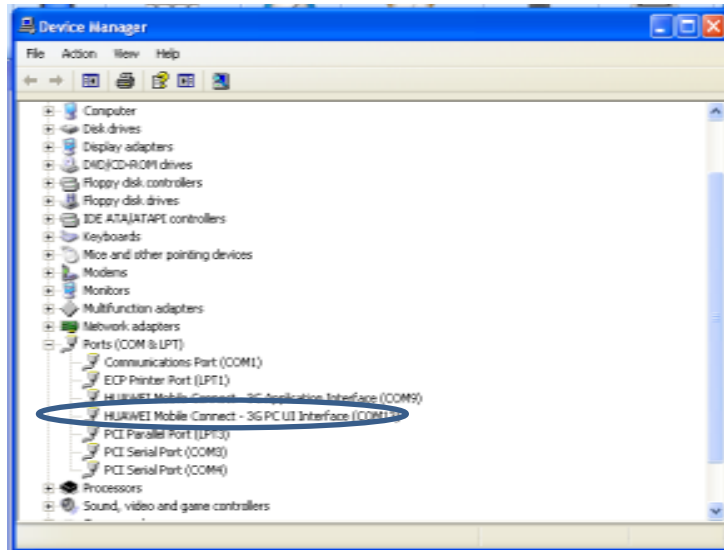


Figure A.5 Device Manager

After verify the comport number to use, select the comport number in the drop down list. Also you need to select your preferred baud rate. After the selection, click connect.

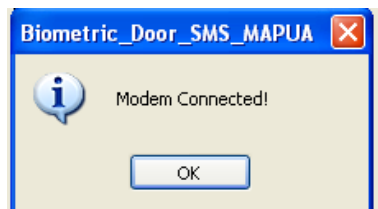


Figure A.6 Modem Connection Successful

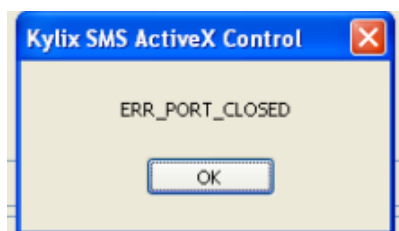


Figure A.7 Modem Connection Error

Figure A.6 shows the message that will be displayed if the attempt to connect the modem is successful. If the modem is not connected Figure A.7 will be shown.

The screenshot displays the 'Accounts Management' window. It features a search bar at the top with the text 'Search Name: mj'. Below the search bar is a table with the following columns: Reg #, ID Number, Full Name, Address, Gender, Contact, Bday, and Username. The table contains several rows of data, with the row for Reg # 20 highlighted. At the bottom of the window, there are buttons for 'View Individual Logs', 'Add', 'Update', 'Delete', and 'Close'.

Reg #	ID Number	Full Name	Address	Gender	Contact	Bday	Username
21	1234	hr	pkiray	Female	09988888	3/6/2012	N/A
20	10000	Mr. Joseph	1234	Female	0917567201	3/6/2012	hr
19	00006	Mike H.	sampleloc	Male	09175686073	3/6/2012	hr
18	00003	Mike admin	sampleloc	Male	09175686073	3/6/2012	mike
16	4	mike helper	qc	Male	09175686073	3/6/2012	N/A
15	00002	Nico	Sampleloc	Male	09364898413	3/6/2012	nico
13	00000	MJ	QC	Female	0916402790	3/6/2012	mj
12	3	q	q	Male	44444444444444	3/6/2012	N/A
11	2	w	w	Male	33333333333333	3/6/2012	N/A
6	000000000	admin	system	Male	00000000000000	3/3/2012	admin

Figure A.8 Accounts Management Interface

This is the form where the admin will add, delete, update and search records. The data displayed by default will be the latest records added in the database.

The screenshot displays the 'Registration Form' window. It includes a red instruction at the top: 'Fill up the Form Completely and Correctly'. The form contains the following fields: Account Type (dropdown), ID Number (text), Full Name (text), Address (text), Gender (dropdown), Contact Number (text), SMS Notification (checkbox labeled 'Enabled SMS Notifications'), Birthday (date picker), Username (text), and Password (text). At the bottom, there are buttons for 'Save', 'Update', and 'Cancel'.

Figure A.9 User Registration Form

When you click the Add button in the accounts management interface, the form in Figure A.7 will be displayed. Fill up the form with the required fields and click Save.

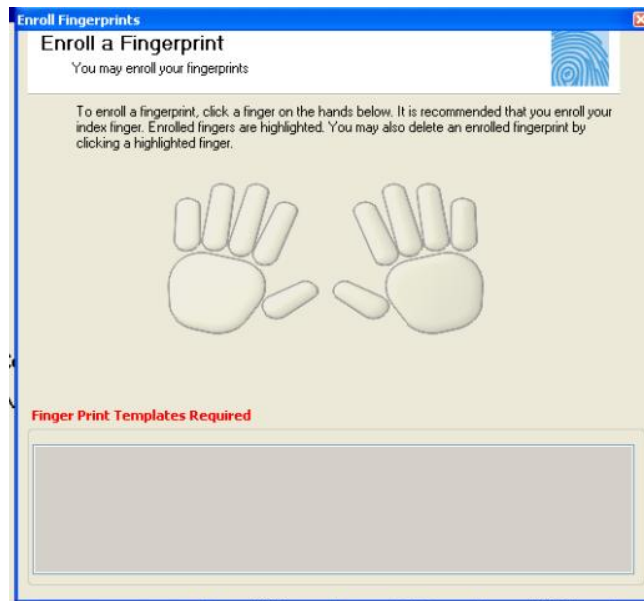


Figure A.10 Fingerprint Template Registration

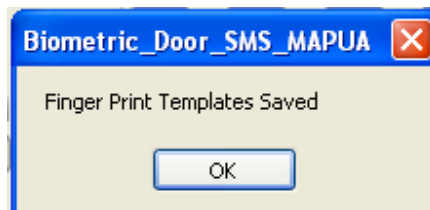


Figure A.11 Fingerprint Template Registration Successful

After clicking Save, you will be redirected to the fingerprint template registration form as shown in Figure A.10. Follow the steps displayed on the screen and if the template registration is successful, Figure A.11 will be displayed.



Figure A.12 Record Selection Error



Figure A.13 Update Successful

If the user wants to update his/her record, select first the record that will be updated and then click the Update button in the Accounts Management form. If no record is selected, Figure A.12 will be shown. Otherwise, a form similar to Figure A.9 will be presented. Edit the fields that need to be updated and click then click the Update button. If update is successful, Figure A.13 will appear.

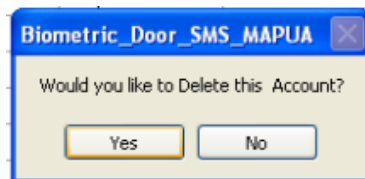


Figure A.14 Delete Verification

To delete a record, click the desired user profile to be removed and then click Delete. A message as shown in Figure A.14 will be displayed to verify if you want to continue with the process. Deleting a record is also done in the Accounts Management Form. If successful, Figure A.15 will be displayed.



Figure A.15 Delete Successful

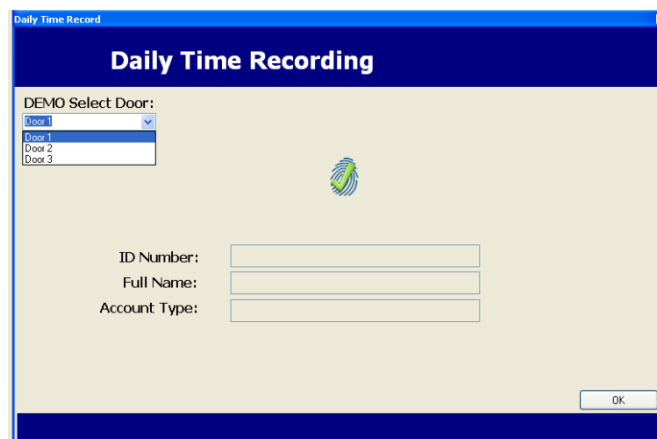


Figure A.16 DTR Form

Figure A.16 shows the DTR form. This should be the form that will be displayed if a fingerprint is to be scanned.

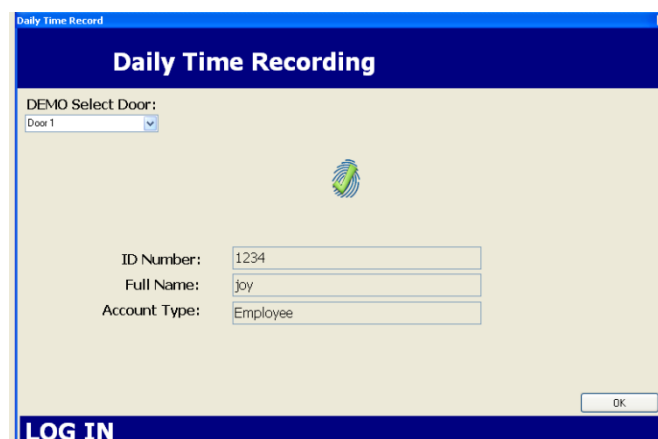




Figure A.17 Log In Successful

Figure A.18 Log Out Successful

Figure A.19 Permission Denied

Whenever the attempt is successful, a confirmation is seen at the bottom part of the interface as shown in Figures A.17 and A.18. However, if the user is not permitted to enter that area, the message as shown in Figure A.19 is displayed.

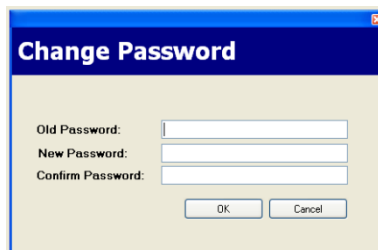
A Windows-style dialog box titled "Change Password" with a blue header bar. The main area has a light beige background. It contains three text input fields labeled "Old Password:", "New Password:", and "Confirm Password:". Below the fields are two buttons: "OK" and "Cancel".

Figure A.20 Change Password

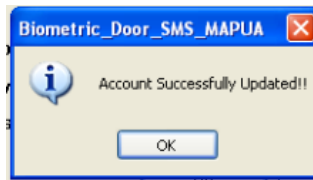


Figure A.21 Change Password Successful

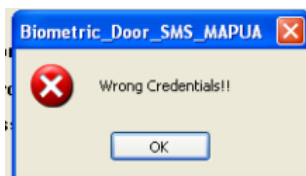


Figure A.22 Change Password Error

If the admin, owner, or HR wants to change his/her current password, he/she may do so by clicking the Change Password tab in the main interface. A form similar to Figure A.20 will appear to complete the procedure. If the

changing of the password is successful, Figure A.21 will be shown; otherwise the user will see Figure A.22.

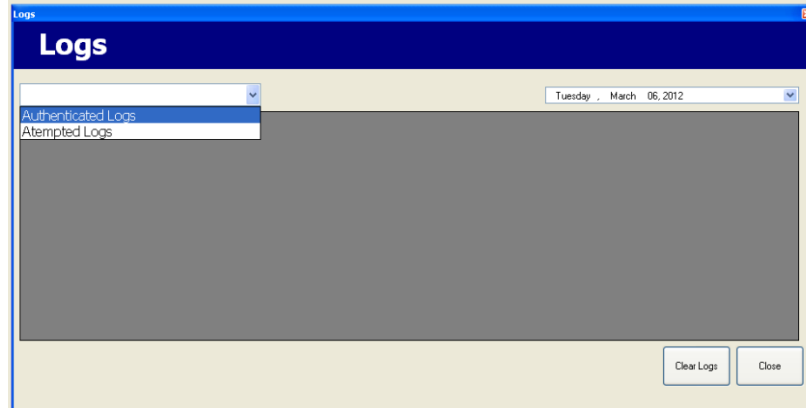


Figure A.23 DTR Logs

Figure A.23 shows the form for the DTR Logs. In this form, you will be able to see all the attempts made using the biometric scanner. There are two options available, authenticated and attempted logs. Authenticated logs will display all the successful attempts. The attempted logs on the other hand will show the attempts of the user to enter an area that they are not permitted.

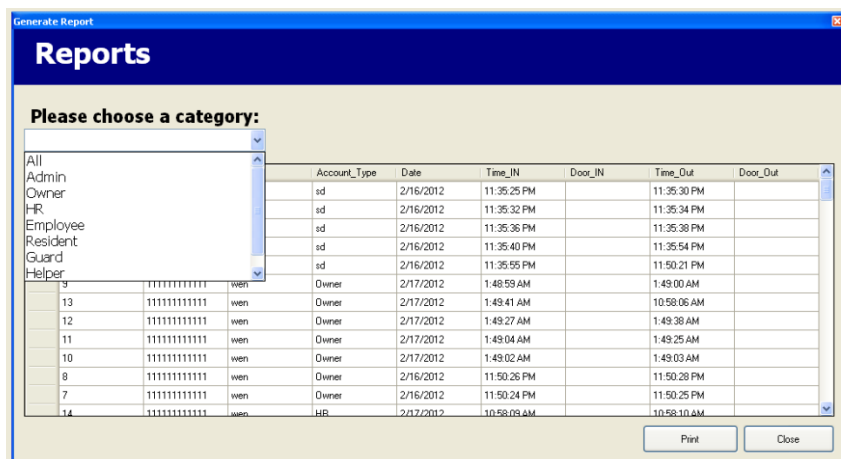


Figure A.24 Report Generation

Row ID	Row Number	Name	Account Type	Date	Date_1	Date_2	Date_3	Date_4	Date_5	Date_6
1	1	ad	ad	20/06/2012	11:00:00 PM					
2	2	ad	ad	20/06/2012	11:00:00 PM					
3	3	ad	ad	20/06/2012	11:00:00 PM					
4	4	ad	ad	20/06/2012	11:00:00 PM					
5	5	ad	ad	20/06/2012	11:00:00 PM					
6	6	ad	ad	20/06/2012	11:00:00 PM					
7	7	ad	ad	20/06/2012	11:00:00 PM					
8	8	ad	ad	20/06/2012	11:00:00 PM					
9	9	ad	ad	20/06/2012	11:00:00 PM					
10	10	ad	ad	20/06/2012	11:00:00 PM					
11	11	ad	ad	20/06/2012	11:00:00 PM					
12	12	ad	ad	20/06/2012	11:00:00 PM					
13	13	ad	ad	20/06/2012	11:00:00 PM					
14	14	ad	ad	20/06/2012	11:00:00 PM					
15	15	ad	ad	20/06/2012	11:00:00 PM					
16	16	ad	ad	20/06/2012	11:00:00 PM					
17	17	ad	ad	20/06/2012	11:00:00 PM					
18	18	ad	ad	20/06/2012	11:00:00 PM					
19	19	ad	ad	20/06/2012	11:00:00 PM					
20	20	ad	ad	20/06/2012	11:00:00 PM					
21	21	ad	ad	20/06/2012	11:00:00 PM					
22	22	ad	ad	20/06/2012	11:00:00 PM					
23	23	ad	ad	20/06/2012	11:00:00 PM					
24	24	ad	ad	20/06/2012	11:00:00 PM					
25	25	ad	ad	20/06/2012	11:00:00 PM					

Figure A.25 Report Generation

To be able to generate a report that consist the logs, click the generate reports button seen on the main interface. Before printing, you have the option to choose what user account type log/s will be printed. Click the desired account type and then click print. A print preview shown in Figure A.25 will be displayed as a sample of the output.

4. Troubleshooting Guides and Procedures

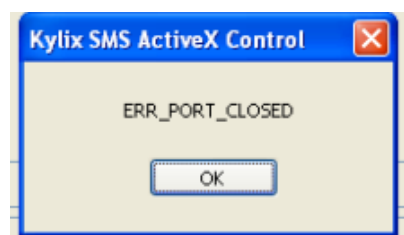


Figure A.26 Modem Not Connected

Figure A.26 shows the error message when the modem is not detected. The problem maybe is that the modem is not connected, or the comport used is incorrect.

Solution

1. Connect the modem

This device is the one responsible for sending the alert messages. Connect the modem to the USB port of the computer.

2. Incorrect comport number

To check if the correct comport number is used go to My Computer>Properties>Device Manager>Ports then search for the port used by the modem. Use this comport number to connect the modem.

Appendix B

Pictures of Prototype

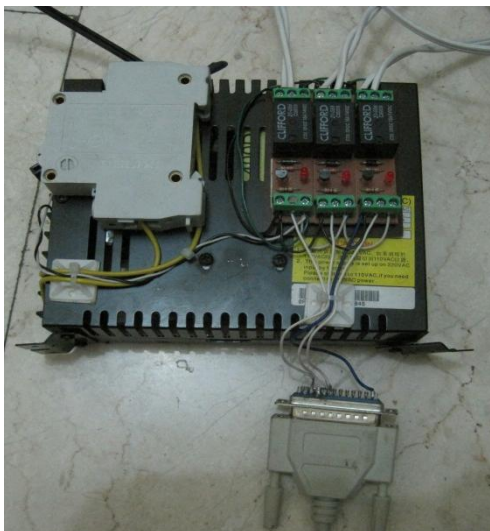


Figure B.1 Power Supply, Relay Driver and Parallel Port



Figure B.2 Mini-Door



Figure B.3 Fingerprint Scanner



Figure B.4 Modem/Broadband

Appendix C

```
Imports MySql.Data.MySqlClient
Public Class frmLogin

    Private Sub frmLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.CboLevel.SelectedIndex = 0
        Me.UsernameTextBox.Focus()
    End Sub
    Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click
        If Me.UsernameTextBox.Text = "" Or Me.PasswordTextBox.Text = "" Then
            MsgBox("Fields required!!!")

        Exit Sub

    End If

    Try

        ' Load From DB
        GlobalFunctions.db_connect()

        Dim reader As MySqlDataReader
```

```

        Dim command As MySqlCommand = connection.CreateCommand()
        'Admin
        If CboLevel.SelectedIndex = 0 Then
            sql = "SELECT * from account_records where Username='" &
Me.UsernameTextBox.Text & "' AND Password='" & Me.PasswordTextBox.Text & "' AND
account_type='Admin'"
        End If
        'Owner
        If CboLevel.SelectedIndex = 1 Then
            sql = "SELECT * from account_records where Username='" &
Me.UsernameTextBox.Text & "' AND Password='" & Me.PasswordTextBox.Text & "' AND
account_type='Owner'"
        End If
        'HR
        If CboLevel.SelectedIndex = 2 Then
            sql = "SELECT * from account_records where Username='" &
Me.UsernameTextBox.Text & "' AND Password='" & Me.PasswordTextBox.Text & "' AND
account_type='HR'"
        End If

        command.CommandText = sql
        reader = command.ExecuteReader()
        If (reader.HasRows) Then
            While (reader.Read())

                MsgBox("Access Granted!!!!", MsgBoxStyle.Information)
                system_user_id = reader("detail_id").ToString
                ' system_u = reader("Name").ToString
                account_type = reader("account_type").ToString

                ,

                Me.Hide()
                Call LOG_USER_ACCOUNT()
                With frmMain
                    .Show()

                End With

                Me.UsernameTextBox.Text = ""
                Me.PasswordTextBox.Text = ""

            End While

            GlobalFunctions.connection.Close()
        Else
            MsgBox("Access Denied!!!!")
            Me.PasswordTextBox.Text = ""
            Me.UsernameTextBox.Text = ""

        End If
    Catch ex As MySqlException
        MessageBox.Show("Error in MySQL Server! Error: " & ex.Message, "",
        MessageBoxButtons.OK, MessageBoxIcon.Error)

```

```

        End Try
    End Sub
    Public Sub LOG_USER_ACCOUNT()

        'If admin
        If account_type = "Admin" Then
            With frmMain
                .ToolStripSMS.Enabled = True
                .ToolStripRun.Enabled = True
                .ToolStripChangePass.Enabled = True
                frmAccounts.btnAdd.Enabled = True
                frmAccounts.CmdDelete.Enabled = True
                frmAccounts.cmdUpdate.Enabled = True
                frmLogs.Button2.Enabled = True
            End With
        End If

        'If hr
        If account_type = "HR" Then
            With frmMain
                .ToolStripAcctMan.Enabled = False
                .ToolStripRun.Enabled = False
                .ToolStripSMS.Enabled = False
                .ToolStripLogs.Enabled = False
                .ToolStripChangePass.Enabled = True
                frmAccounts.btnAdd.Enabled = False
                frmAccounts.CmdDelete.Enabled = False
                frmAccounts.cmdUpdate.Enabled = False
            End With
        End If
        'if owner
        If account_type = "Owner" Then
            With frmMain
                .ToolStripAcctMan.Enabled = False
                .ToolStripSMS.Enabled = False
                .ToolStripRun.Enabled = False
                .ToolStripChangePass.Enabled = True
                frmAccounts.btnAdd.Enabled = False
                frmAccounts.CmdDelete.Enabled = False
                frmAccounts.cmdUpdate.Enabled = False
                frmLogs.Button2.Enabled = False
            End With
        End If
    End Sub

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
        Me.PasswordTextBox.Text = ""
        Me.UsernameTextBox.Text = ""
        Me.Close()
    End Sub

```

```

        End Sub

        Private Sub Label16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label16.Click

        End Sub
    End Class

Public Class frmMain

        Private Sub ToolStripAcctMan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripAcctMan.Click
            frmAccounts.ShowDialog()
        End Sub

        Private Sub ToolStripPlantMgt_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripRun.Click
            frmDTR.ShowDialog()
        End Sub

        Private Sub ToolStripQuit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripQuit.Click
            Dim ans As String

            ans = MsgBox("Would you like to Quit?", vbYesNo)
            If ans = vbYes Then

                Me.AxKylixSMS1.Disconnect()
            End
            Else

            End If
        End Sub

        Private Sub ToolStripHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripHelp.Click
            frmHelp.ShowDialog()
        End Sub

        Private Sub ToolStripLogs_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripLogs.Click
            Dim frmlogs As New frmLogs
            frmLogs.ShowDialog()
        End Sub

        Private Sub ToolStripReport_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripReport.Click
            frmReports.ShowDialog()
        End Sub

        Private Sub ToolStripLogOut_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripLogOut.Click
            Me.AxKylixSMS1.Disconnect()
            frmLogin.Show()
            Me.Close()
        End Sub
    End Class

```

```

End Sub

Public Sub SEND_SMS(ByVal mobilenumber As String, ByVal sms As String)
    With AxKylixSMS1
        Dim Reference As Integer

        .SendRetryTimes = 2

        Reference = .SendSMS(mobilenumber, sms)

        If Reference < 1 Then
            .GetLastError(1)
        Else
            ' MsgBox("Successful." & vbCrLf & "Reference: " & Reference)
        End If
    End With
End Sub

Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripSMS.Click
    FrmModemSettings.ShowDialog()
End Sub

Private Sub frmMain_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
    Me.AxKylixSMS1.Disconnect()
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub

Private Sub ToolStripChangePass_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ToolStripChangePass.Click
    frmChange.ShowDialog()
End Sub

Private Sub frmMain_Load_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub
End Class

Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports

Public Class FrmModemSettings

    Dim myPort As Array 'COM Ports detected on the system will be stored here

```



```

Private Sub FrmModemSettings_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'Display Ports list
    'When our form loads, auto detect all serial ports in the system and
    populate the cmbPort Combo box.
    myPort = IO.Ports.SerialPort.GetPortNames() 'Get all com ports available

    For i = 0 To UBound(myPort)
        CboPorts.Items.Add(myPort(i))
    Next

    'Displays Baudrate list
    With Me.CboBaudrate
        .Items.Add("9600")
        .Items.Add("19200")
        .Items.Add("38400")
        .Items.Add("57600")
        .Items.Add("115200")
        .SelectedIndex = 0
    End With

    Me.CboPorts.SelectedIndex = 0
    Me.CboBaudrate.SelectedIndex = 0

End Sub

Private Sub BtnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConnect.Click

    If BtnConnect.Text = "&Connect" Then
        If Me.CboPorts.Text = "" Then
            MsgBox("Please choose comport!!")
            Exit Sub
        End If

        If Me.CboBaudrate.Text = "" Then
            MsgBox("Please choose Baudrate")
            Exit Sub
        End If

        With frmMain.AxKylixSMS1

            .RegisterName = "5265796D6F6E642046657264696E616E"
            .RegisterCode =
"21A75BE04BF69B5C45180020D71A7E14A0D539004ACE1F047D8C11908E41DA26EB7D62F11BBF9B4F"

            .NeedLog = 1 'If you have any problem please set the value to
1, and send us the log file "C:\KylixSMS.LOG".
            ' .PINCode = PINCode
            .ConnectionMode = 1
            .ConnectionProtocol = 1
            .AutoDeleteNewSMS = 1
            .AutoDeleteAllReport = 1
            .ConnectionParameter = Me.CboPorts.Text + ", " +
Me.CboBaudrate.Text

```

```

        If .Connect < 1 Then
            .GetLastError(1)
        Else
            BtnConnect.Text = "&Disconnect"

            MsgBox("Modem Connected!", MsgBoxStyle.Information)
            ' isconnected = True
            ' Main.p.Visible = True
            ' Main.toolGSMSignal.Text = Main.toolGSMSignal.Text & " " &
            .GetSignalLevel()

            Me.Hide()
        End If
    End With

```

```

Else
    frmMain.AxKylixSMS1.Disconnect()

    BtnConnect.Text = "&Connect"
    'Main.picSignal.Visible = True
    MsgBox("Modem Disconnected!", MsgBoxStyle.Information)
    ' isconnected = False

    End If

```

End Sub

```

Private Sub cmdOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdOK.Click
    Me.Hide()

```

End Sub
End Class

```

Imports MySql.Data.MySqlClient
Public Class frmAccounts

```

```

    Private Sub frmAccounts_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        LOAD_ACCOUNT_RECORDS()
    End Sub

```

```

    Public Sub LOAD_ACCOUNT_RECORDS()
        ' Connect to Database
        GlobalFunctions.db_connect()
        Try
            Dim Adapter As New MySqlDataAdapter("SELECT * FROM account_records
ORDER BY detail_id DESC", GlobalFunctions.MySQLConnectionString)
            Dim dt As New DataTable("accounts")
            Adapter.Fill(dt)
            Me.DataGridView1.DataSource = dt

            With Me.DataGridView1

```

```

        .Columns(0).HeaderText = "Reg #"
        .Columns(1).HeaderText = "ID Number"
        .Columns(2).HeaderText = "Full Name"
        .Columns(3).HeaderText = "Address"
        .Columns(4).HeaderText = "Gender"
        .Columns(5).HeaderText = "Contact"
        .Columns(6).HeaderText = "Bday"

        .Columns(7).HeaderText = "Username"
        .Columns(8).HeaderText = "Password"
        .Columns(9).HeaderText = "Account_Type"
        .Columns(10).HeaderText = "SMS Notify"
        .Columns(8).Width = 0
    End With

    Catch ex As MySqlException
        MessageBox.Show("Error " & ex.ToString, "Loading Records")
    End Try
    'close connections
    GlobalFunctions.connection.Close()

End Sub

Public Sub SEARCH_ACCOUNT_RECORDS()
    ' Connect to Database
    GlobalFunctions.db_connect()
    Try
        Dim Adapter As New MySqlDataAdapter("SELECT * FROM account_records
WHERE Name like '%" & Me.txtSearch.Text & "%' ORDER BY detail_id DESC",
GlobalFunctions.MySQLConnectionString)
        Dim dt As New DataTable("accounts")
        Adapter.Fill(dt)
        Me.DataGridView1.DataSource = dt

        With Me.DataGridView1
            .Columns(0).HeaderText = "Reg #"
            .Columns(1).HeaderText = "ID Number"
            .Columns(2).HeaderText = "Full Name"
            .Columns(3).HeaderText = "Gender"
            .Columns(4).HeaderText = "Contact"
            .Columns(5).HeaderText = "Birthday"
            .Columns(6).HeaderText = "Username"
            .Columns(7).HeaderText = "Password"
            .Columns(8).HeaderText = "Account Type"
        End With

        Catch ex As MySqlException
            MessageBox.Show("Error " & ex.ToString, "Loading Records")
        End Try
        'close connections
        GlobalFunctions.connection.Close()

    End Sub

    Private Sub txtSearch_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtSearch.KeyPress
        Call SEARCH_ACCOUNT_RECORDS()
    End Sub

```

```

    Private Sub txtSearch_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtSearch.TextChanged
        Call SEARCH_ACCOUNT_RECORDS()
    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        strid = ""
        ClearTextBox(Me)
        Me.Close()
    End Sub

    Private Sub CmdDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CmdDelete.Click
        If strid = "" Or detail_id = "" Then
            MsgBox("Please choose record first!!", MsgBoxStyle.Critical)

            Exit Sub
        End If

        Dim ans As String

        ans = MsgBox("Would you like to Delete this Account?", vbYesNo)
        If ans = vbYes Then

            Call DELETE_ACCOUNT()
        Else

            End If
        End Sub

        Private Sub cmdUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUpdate.Click
            If strid = "" Then
                MsgBox("Please choose record first!!", MsgBoxStyle.Critical)

                Exit Sub
            End If

            If DataGridView1.ColumnCount > 1 Then

                On Error Resume Next

                strid = DataGridView1.CurrentRow.Cells(0).Value.ToString()

                ' Load From DB
                GlobalFunctions.db_connect()

                Dim reader As MySqlDataReader

                Dim command As MySqlCommand = connection.CreateCommand()
                command.CommandText = "SELECT * FROM account_records WHERE
detail_id='" & strid & "'"
                reader = command.ExecuteReader()
                If (reader.HasRows) Then
                    While (reader.Read())

```

```

        With frmRegisterAccount

            .txtID.Text = reader(1).ToString
            .txtname.Text = reader(2).ToString
            .TxtAddress.Text = reader(3).ToString
            .Cbogender.Text = reader(4).ToString
            .Txtcontact.Text = reader(5).ToString
            .DateTimePicker1.Value = reader(6).ToString
            .txtusername.Text = reader(7).ToString
            .txtPassword.Text = reader(8).ToString

            .CboAcctType.Text = reader(9).ToString
            .checkSMS.Checked = reader(10).ToString
            .cmdUpdate.Enabled = True
            .cmdSave.Enabled = False
            .ShowDialog()

        End With

    End While

    GlobalFunctions.connection.Close()

Else

    End If
End If
End Sub
Public Sub DELETE_ACCOUNT()

    ' Connect to Database
    GlobalFunctions.db_connect()
    Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()
    Try
        command = New MySqlCommand("DELETE FROM account_records WHERE
detail_id='" & detail_id & "'", GlobalFunctions.connection, transaction)
        command.ExecuteNonQuery()

        command = New MySqlCommand("DELETE FROM finger_template WHERE
Account_ID='" & strid & "'", GlobalFunctions.connection, transaction)
        command.ExecuteNonQuery()
        transaction.Commit()
        MsgBox("Record Deleted!!!")
        Call LOAD_ACCOUNT_RECORDS()

        strid = ""
        detail_id = ""

    Catch ex As MySqlException
        MessageBox.Show("Error in DELETING data! Error: " & ex.Message, "Data
Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        transaction.Rollback()
    End Try
    'close connections
    GlobalFunctions.connection.Close()

```

```

End Sub

Private Sub DataGridView1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles DataGridView1.Click
    If DataGridView1.ColumnCount > 1 Then
        On Error Resume Next

        strid = DataGridView1.CurrentRow.Cells(1).Value.ToString()
        detail_id = DataGridView1.CurrentRow.Cells(0).Value.ToString()

        CmdDelete.Enabled = True
        cmdUpdate.Enabled = True
    Else
        ' End If

    End If
End Sub

Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
    With frmRegisterAccount
        .cmdUpdate.Enabled = False

        .cmdSave.Enabled = True
        .ShowDialog()

    End With
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If strid = "" Then
        MsgBox("Please choose record first!!", MsgBoxStyle.Critical)

        Exit Sub
    End If

    If DataGridView1.ColumnCount > 1 Then

        On Error Resume Next

        strid = DataGridView1.CurrentRow.Cells(0).Value.ToString()

        ' Load From DB
        GlobalFunctions.db_connect()

        Dim reader As MySqlDataReader

        Dim command As MySqlCommand = connection.CreateCommand()
        command.CommandText = "SELECT * FROM account_records WHERE
detail_id='" & strid & "'"
        reader = command.ExecuteReader()
        If (reader.HasRows) Then
            While (reader.Read())

```

```

        With frmIndDTR

            .txtID.Text = reader(1).ToString
            .txtname.Text = reader(2).ToString
            .txtAccountType.Text = reader(9).ToString

            .ShowDialog()

        End With

    End While

    GlobalFunctions.connection.Close()

Else

    End If
End If
End Sub

Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
DataGridView1.CellContentClick

    End Sub

Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click

    End Sub
End Class

Imports System.Windows.Forms
Imports MySql.Data.MySqlClient
Public Class frmRegisterAccount

    'check duplicated values
    Sub if_EXIST()
        ' Load From DB
        GlobalFunctions.db_connect()

        Dim reader As MySqlDataReader

        Dim command As MySqlCommand = connection.CreateCommand()
        command.CommandText = "SELECT * from account_records where ID_number='" &
Me.txtID.Text & "'"
        reader = command.ExecuteReader()
        If (reader.HasRows) Then
            While (reader.Read())
                MsgBox("ID Number already issued!!, Please input another!",
MsgBoxStyle.Exclamation)
            End While
        End If
    End Sub

```

```

        GlobalFunctions.connection.Close()
        Me.txtID.Text = ""
        Me.txtID.Focus()
    Else
        'call add account function

        ADD_ACCOUNT()
    End If

End Sub
Sub ADD_ACCOUNT()

    ' Connect to Database
    GlobalFunctions.db_connect()
    Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()
    Try
        'call sql insert command
        command = New MySqlCommand("INSERT INTO account_records values('','" &
txtID.Text & "',''" & txtname.Text & "',''" & TxtAddress.Text & "',''" &
Me.Cbogender.Text & "',''" & Me.Txtcontact.Text & "',''" &
Me.DateTimePicker1.Value.Date & "',''" & Me.txtusername.Text & "',''" &
Me.txtPassword.Text & "',''" & Me.CboAcctType.Text & "',''" & Me.checkSMS.Checked &
"'') ", GlobalFunctions.connection, transaction)
        ' command.Parameters.Add(New MySqlParameter("image", imgBytes))
        command.ExecuteNonQuery()
        transaction.Commit()
        strid = txtID.Text 'variable for enrolling Fprint

        ClearTextBox(Me) 'clear textboxes
        Dim f As New frmEnrollFinger
        With f

            f.ShowDialog()

        End With

        Catch ex As MySqlException
            MessageBox.Show("Error in inserting new data! Error: " & ex.Message,
            "Data Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            transaction.Rollback()
        End Try
        'close connections
        GlobalFunctions.connection.Close()
    End Sub

    Private Sub cmdCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCancel.Click
        ClearTextBox(Me)
        Me.Close()

    End Sub

```



```

Private Sub cmdSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSave.Click
    If txtID.Text = "" Or Me.txtname.Text = "" Or Me.Cbogender.Text = "" Or
TxtAddress.Text = "" Or Me.Txttcontact.Text = "" Or CboAcctType.Text = "" Or
txtusername.Text = "" Or Me.txtPassword.Text = "" Then
        MsgBox("Pls. Complete Entry!", vbExclamation, "")
        Exit Sub
    End If

    If IsNumeric(Me.Txttcontact.Text) = False Then

        MsgBox("Numeric Only!!", vbCritical, "error")
        Me.Txttcontact.Text = ""
        Me.Txttcontact.Focus()

        Exit Sub
    End If

    'check if id is existing the database
    If Me.txtID.Text = "N/A" Then
        ADD_ACCOUNT()
    Else

        if_EXIST()
    End If

End Sub
Public Sub UPDATE_STUDENT_ACCOUNT()
    ' Connect to Database
    GlobalFunctions.db_connect()
    ' Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()

    Try
        GlobalFunctions.execute_nonquery("Update account_records set
Id_Number='" & Me.txtID.Text & "',Name='" & Me.txtname.Text & "',address='" &
Me.TxtAddress.Text & "',Gender='" & Me.Cbogender.Text & "',contact='" &
Me.Txttcontact.Text & "',bday='" & Me.DateTimePicker1.Value.Date & "',username='" &
Me.txtusername.Text & "',password='" & Me.txtPassword.Text & "',account_type='" &
Me.CboAcctType.Text & "',sms_notify='" & Me.checkSMS.Checked & "'" WHERE
Detail_ID='" & strid & "'")

        transaction.Commit()

        MsgBox("Successfully Updated!!", MsgBoxStyle.Information)
        frmAccounts.LOAD_ACCOUNT_RECORDS()

        strid = ""
        ClearTextBox(Me)
        Me.Close()

    Catch ex As MySqlException
        MessageBox.Show("Error in Updating data! Error: " & ex.Message, "",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
        transaction.Rollback()
    End Try
End Sub

```

```

        End Try
        ' close connections
        GlobalFunctions.connection.Close()

    End Sub

    Private Sub cmdUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUpdate.Click
        CheckMyTextBox(Me)
        If txtID.Text = "" Or Me.txtname.Text = "" Or Me.Cbogender.Text = "" Or
TxtAddress.Text = "" Or Me.Txtcontact.Text = "" Or CboAcctType.Text = "" Or
txtusername.Text = "" Or Me.txtPassword.Text = "" Then
            MsgBox("Pls. Complete Entry!", vbExclamation, "")
            Exit Sub
        End If

        If IsNumeric(Me.Txtcontact.Text) = False Then

            MsgBox("Numeric Only!!", vbCritical, "error")
            Me.Txtcontact.Text = ""
            Me.Txtcontact.Focus()

            Exit Sub
        End If

        UPDATE_STUDENT_ACCOUNT()

        'End If
    End Sub

    Private Sub CboAcctType_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CboAcctType.SelectedIndexChanged
        GetAutoID()
        With Me
            .txtID.ReadOnly = False
            .txtusername.ReadOnly = False
            .txtPassword.ReadOnly = False
            .txtID.Text = ""
            .txtusername.Text = ""
            .txtPassword.Text = ""
        End With

        If CboAcctType.SelectedIndex = 3 Then
            With Me
                .txtusername.ReadOnly = True
                .txtPassword.ReadOnly = True
                .txtusername.Text = "N/A"
                .txtPassword.Text = "N/A"
            End With
        End If

        If CboAcctType.SelectedIndex = 4 Then

```

```

        GetAutoID()
        With Me
            ' .txtID.Text = "N/A"
            .txtID.ReadOnly = True
            .txtusername.ReadOnly = True
            .txtPassword.ReadOnly = True
            .txtusername.Text = "N/A"
            .txtPassword.Text = "N/A"
        End With
    End If

    If CboAcctType.SelectedIndex = 5 Then
        With Me
            .txtusername.ReadOnly = True
            .txtPassword.ReadOnly = True
            .txtusername.Text = "N/A"
            .txtPassword.Text = "N/A"
        End With
    End If
    If CboAcctType.SelectedIndex = 6 Then

        GetAutoID()
        With Me
            ' .txtID.Text = "N/A"
            .txtID.ReadOnly = True
            .txtusername.ReadOnly = True
            .txtPassword.ReadOnly = True
            .txtusername.Text = "N/A"
            .txtPassword.Text = "N/A"
        End With
    End If
    If CboAcctType.SelectedIndex = 7 Then
        With Me
            ' .txtID.Text = "N/A"
            ' .txtID.ReadOnly = True
            .txtusername.ReadOnly = True
            .txtPassword.ReadOnly = True
            .txtusername.Text = "N/A"
            .txtPassword.Text = "N/A"
        End With
    End If

End Sub

Sub GetAutoID()

    ' Load From DB
    GlobalFunctions.db_connect()

    Dim reader As MySqlDataReader
    Dim command As MySqlCommand = connection.CreateCommand()
    '
    command.CommandText = "SELECT MAX(ID_number+1) as ID_Number from
account_records where account_type='Helper' or account_type='Resident'"
    reader = command.ExecuteReader()

```

```

        If (reader.HasRows) Then
            While (reader.Read())
                Me.txtID.Text = reader("ID_Number").ToString
            Exit While
        End While
    Else

        Me.txtID.Text = "1"
    End If

    GlobalFunctions.connection.Close()

End Sub

Private Sub frmRegisterAccount_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub

Private Sub Label15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label15.Click

End Sub
End Class

Imports MySql.Data.MySqlClient
Imports DPFP
Public Class frmEnrollFinger
    Dim conn As MySqlConnection

    Private Sub EnrollmentControl_OnEnroll(ByVal Control As Object, ByVal
FingerMask As Integer, ByVal Template As DPFP.Template, ByRef EventHandlerStatus
As DPFP.Gui.EventHandlerStatus) Handles EnrollmentControl.OnEnroll

    Try

        Dim reader As MySqlDataReader
        conn = New MySqlConnection()
        Dim bytes As Byte() = Nothing
        Dim command As MySqlCommand = conn.CreateCommand()
        conn.ConnectionString = ConnectionString.ConnString
        command.CommandText = "Select * from finger_template where
Account_ID='" & strid & "' and Fingermask = '" & FingerMask & "'"
        Try
            conn.Open()
            reader = command.ExecuteReader

            If (reader.HasRows) Then
                conn.Close()
                conn.Open()
                Template.Serialize(bytes)
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub
End Class

```

```

        Dim Str As String = "UPDATE finger_template SET byte_template
= ?ImageData WHERE (Account_ID = '" & strid & "') and (Fingermask = '" &
FingerMask & "')"
        Dim cmd As MySqlCommand = New MySqlCommand(Str, conn)
        cmd.Parameters.Add(New MySqlParameter("?ImageData", bytes))
        Try

            Dim icount As Integer
            icount = cmd.ExecuteNonQuery

            MsgBox("Finger Print Templates Updated")
            conn.Close()

            strid = ""
            Me.Close()
            frmRegisterAccount.Close()
            frmAccounts.LOAD_ACCOUNT_RECORDS()

        Catch ex As Exception
            MsgBox(ex.Message)
            MsgBox("Theres an Error")
        End Try

    Else

        Template.Serialize(bytes)
        conn.Close()
        conn.Open()
        Dim Str As String = "INSERT INTO finger_template
(byte_template, Account_ID,Fingermask) values (?ImageData, '" & strid & "', '" &
FingerMask & "')"
        Dim cmd As MySqlCommand = New MySqlCommand(Str, conn)
        cmd.Parameters.Add(New MySqlParameter("?ImageData", bytes))

        Try

            Dim icount As Integer
            icount = cmd.ExecuteNonQuery

            MsgBox("Finger Print Templates Saved")
            conn.Close()

            strid = ""
            Me.Close()
            frmRegisterAccount.Close()
            frmAccounts.LOAD_ACCOUNT_RECORDS()

        Catch ex As Exception
            MsgBox(ex.Message)

        End Try

    End If

```

```

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Private Sub EnrollmentControl_OnDelete(ByVal Control As Object, ByVal
FingerMask As Integer, ByRef EventHandlerStatus As DPFP.Gui.EventHandlerStatus)
Handles EnrollmentControl.OnDelete
        'MsgBox("Can't do changes in this part!", MsgBoxStyle.Exclamation, "Cannot
Delete...")
        '    EventHandlerStatus = Gui.EventHandlerStatus.Failure
    End Sub

    Private Sub EnrollmentControl_OnCancelEnroll(ByVal Control As System.Object,
ByVal ReaderSerialNumber As System.String, ByVal Finger As System.Int32) Handles
EnrollmentControl.OnCancelEnroll
        ListEvents.Items.Insert(0, String.Format("OnCancelEnroll: {0}, finger
{1}", ReaderSerialNumber, Finger))
    End Sub

    Private Sub EnrollmentControl_OnComplete(ByVal Control As System.Object, ByVal
ReaderSerialNumber As System.String, ByVal Finger As System.Int32) Handles
EnrollmentControl.OnComplete
        ListEvents.Items.Insert(0, String.Format("OnComplete: {0}, finger {1}",
ReaderSerialNumber, Finger))
    End Sub

    Private Sub EnrollmentControl_OnFingerRemove(ByVal Control As System.Object,
ByVal ReaderSerialNumber As System.String, ByVal Finger As System.Int32) Handles
EnrollmentControl.OnFingerRemove
        ListEvents.Items.Insert(0, String.Format("OnFingerRemove: {0}, finger
{1}", ReaderSerialNumber, Finger))
    End Sub

    Private Sub EnrollmentControl_OnFingerTouch(ByVal Control As System.Object,
ByVal ReaderSerialNumber As System.String, ByVal Finger As System.Int32) Handles
EnrollmentControl.OnFingerTouch
        ListEvents.Items.Insert(0, String.Format("OnFingerTouch: {0}, finger {1}",
ReaderSerialNumber, Finger))
    End Sub

    Private Sub EnrollmentControl_OnReaderConnect(ByVal Control As System.Object,
ByVal ReaderSerialNumber As System.String, ByVal Finger As System.Int32) Handles
EnrollmentControl.OnReaderConnect
        ListEvents.Items.Insert(0, String.Format("OnReaderConnect: {0}, finger
{1}", ReaderSerialNumber, Finger))

    End Sub

    Private Sub EnrollmentControl_OnReaderDisconnect(ByVal Control As
System.Object, ByVal ReaderSerialNumber As System.String, ByVal Finger As
System.Int32) Handles EnrollmentControl.OnReaderDisconnect
        ListEvents.Items.Insert(0, String.Format("OnReaderDisconnect: {0}, finger
{1}", ReaderSerialNumber, Finger))

```

```

End Sub

Private Sub EnrollmentControl_OnSampleQuality(ByVal Control As System.Object,
ByVal ReaderSerialNumber As System.String, ByVal Finger As System.Int32, ByVal
CaptureFeedback As DPFP.Capture.CaptureFeedback) Handles
EnrollmentControl.OnSampleQuality
    ListEvents.Items.Insert(0, String.Format("OnSampleQuality: {0}, finger
{1}, {2}", ReaderSerialNumber, Finger, CaptureFeedback))
End Sub

Private Sub EnrollmentControl_OnStartEnroll(ByVal Control As System.Object,
ByVal ReaderSerialNumber As System.String, ByVal Finger As System.Int32) Handles
EnrollmentControl.OnStartEnroll
    ListEvents.Items.Insert(0, String.Format("OnStartEnroll: {0}, finger {1}",
ReaderSerialNumber, Finger))
End Sub

Private Sub EnrollmentForm_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ListEvents.Items.Clear()
End Sub

Private Sub frmEnrollFinger_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub
End Class

Option Strict Off
Option Explicit On

Imports MySql.Data.MySqlClient
Imports System.Windows.Forms

Imports DPFP

Module InpOut32_Declarations
    'Inp and Out declarations for port I/O using inpout32.dll.
    Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal
PortAddress As Short) As Short
    Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As
Short, ByVal Value As Short)

End Module
Public Class frmDTR

    'INPOUT32
    Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

    Const default_1 = 128
    Const Pin10 = 192
    Const Pin11 = 0
    Const Pin12 = 160
    Const Pin13 = 144
    Const Pin15 = 136

    'Digital Persona
    Dim counter As Integer = 0

```

```

Dim oConn As New MySqlConnection(ConnectionString.ConnString)

Private matcher As DPFP.Verification.Verification
Private matchResult As DPFP.Verification.Verification.Result

Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click
    Me.Close()
End Sub

Public Sub SEARCH_ACCOUNT()
    Try
        ' Load From DB
        GlobalFunctions.db_connect()

        Dim reader As MySqlDataReader

        Dim command As MySqlCommand = connection.CreateCommand()
        command.CommandText = "SELECT * FROM account_records WHERE
Id_number='" & strid & "'" 'check id number if existing
        reader = command.ExecuteReader()
        If (reader.HasRows) Then
            While (reader.Read())
                With Me
                    'plot the data into controls

                    .txtID.Text = reader(1).ToString
                    .txtname.Text = reader(2).ToString
                    .txtAccountType.Text = reader(9).ToString
                    account_type = reader(9).ToString

                End With

                Call OPEN_DOOR()
                ' Call LOG_ACCOUNT()      'look up if login /log out
                Timer1.Enabled = True

            End While
        Else
            '      Me.lblStatus.Text = "ID not recognized!"

        End If
        Catch ex As Exception
            MessageBox.Show("error scanning: " & ex.Message)
        End Try

        GlobalFunctions.connection.Close()

    End Sub

Public Sub LOG_ACCOUNT()
    Try

```



```

        ' Load From DB
        GlobalFunctions.db_connect()

        Dim reader As MySqlDataReader

        Dim command As MySqlCommand = connection.CreateCommand()

        command.CommandText = "SELECT * FROM account_logs WHERE id_number='" &
strid & "' AND Time_Out='Null' AND Date='" & Date.Today & "'"
        reader = command.ExecuteReader()
        If (reader.HasRows) Then
            While (reader.Read())

                End While

            'logout
            Call ACCOUNT_LOGOUT()

        Else

            'log in
            Call ACCOUNT_LOGIN()

        End If

    Catch ex As Exception
        MessageBox.Show("Error scanning: " & ex.Message)
    End Try
    GlobalFunctions.connection.Close()

End Sub
'insert login data
Public Sub ACCOUNT_LOGIN()
    ' Connect to Database
    GlobalFunctions.db_connect()
    Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()
    Try
        command = New MySqlCommand("INSERT INTO account_logs values('','" &
txtID.Text & "',''" & txtname.Text & "',''" & Me.txtAccountType.Text & "',''" &
Date.Today & "',''" & TimeOfDay & "',''" & Me.cboDoor.Text & "','Null','Null') ",
GlobalFunctions.connection, transaction)
        command.ExecuteNonQuery()
        transaction.Commit()

        Me.lblStatus.Text = "LOG IN" 'set status to login

    Catch ex As MySqlException
        MessageBox.Show("Error in inserting new data! Error: " & ex.Message,
"Data Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        transaction.Rollback()
    End Try
    'close connections
    GlobalFunctions.connection.Close()

```

```

End Sub

Public Sub ACCOUNT_LOGOUT()
    ' Connect to Database
    GlobalFunctions.db_connect()
    ' Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()

    Try
        GlobalFunctions.execute_nonquery("Update account_logs set Time_Out='"
& TimeOfDay & "' ,Door_Out='" & Me.cboDoor.Text & "' WHERE ID_Number='" & strid &
"' AND Time_Out='Null' AND Door_Out='Null' AND Date='" & Date.Today & "'")
        transaction.Commit()
        lblStatus.Text = "LOG OUT" ' set status to logout

    Catch ex As MySqlException
        MessageBox.Show("Error in Updating data! Error: " & ex.Message, "",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
        transaction.Rollback()
    End Try
    ' close connections
    GlobalFunctions.connection.Close()

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    'counter for display
    counter += 1
    If counter = 6 Then

        Call ClearTextBox(Me)
        Me.lblStatus.Text = ""
        ' Lblverify.Text = ""
        Timer1.Enabled = False
        counter = 0

    End If
End Sub

Private Sub frmDTR_Activated(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Activated
    Try
        Me.VerificationControl.Focus()

    Catch ex As MySqlException
        MessageBox.Show("System Error: " & ex.Message, "",
        MessageBoxButtons.OK, MessageBoxIcon.Error)

    End Try

End Sub

Private Sub frmDTR_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    matcher = New Verification.Verification()
    matchResult = New Verification.Verification.Result

```

```

Me.VerificationControl.Focus()
cboDoor.SelectedIndex = 0

' Out(&H378, 0) ' initialized data output to printerport 0000 0000...

End Sub

Private Sub VerificationControl_OnComplete(ByVal Control As Object, ByVal
FeatureSet As DPFP.FeatureSet, ByRef EventHandlerStatus As
DPFP.Gui.EventHandlerStatus) Handles VerificationControl.OnComplete
    Dim strSQL As String = "Select * from finger_template"
    Dim oDa As New MySqlDataAdapter(strSQL, oConn)
    Dim dt As New DataTable
    Dim dr As DataRow
    Try
        oDa.Fill(dt)
        For Each dr In dt.Rows

            'Lblverify.ForeColor = Color.Red
            ' Lblverify.Visible = True
            Dim bytes As Byte() = Nothing
            bytes = dr.Item("byte_template")

            Dim tmlate = New DPFP.Template()
            tmlate.DeSerialize(bytes)
            matcher.Verify(FeatureSet, tmlate, matchResult)

            If matchResult.Verified Then

                EventHandlerStatus = DPFP.Gui.EventHandlerStatus.Success
                strid = dr.Item("Account_ID")

                fail_log = 0 '
                'call search account
                Call SEARCH_ACCOUNT()

                Exit For ' success
            End If

            If Not matchResult.Verified Then EventHandlerStatus =
DPFP.Gui.EventHandlerStatus.Failure
            ' Lblverify.ForeColor = Color.Red
            ' Lblverify.Text = "PLEASE TRY AGAIN..."
            fail_log = fail_log + 1 '
            If fail_log = 10 Then '
                fail_log = 0 'sequence return to zero '
                ' MsgBox("You have used 3 attempt!this window will be
closed!", vbCritical, "-Error") '
                Call SEND_NOTIFICATIONS() 'send sms '

            End If '

            ' MsgBox(fail_log)
            fail_log = fail_log '

```

```

        Timer1.Start()

    Next
    Catch ex As Exception
    End Try
End Sub

Public Sub SEND_NOTIFICATIONS()

    'Door 1
    If cboDoor.SelectedIndex = 0 Then

        sql = "SELECT * from account_records where sms_notify='True' AND
(account_type='Owner' OR account_type = 'Resident' OR account_type = 'Helper' OR
account_type = 'Resident/Employee')"

    End If

    'Door 2
    If cboDoor.SelectedIndex = 1 Then

        sql = "SELECT * from account_records where sms_notify='True' AND
(account_type = 'Admin')"

    End If

    'Door 3
    If cboDoor.SelectedIndex = 2 Then

        sql = "SELECT * from account_records where sms_notify='True' AND
(account_type = 'Owner' Or account_type = 'Resident' Or account_type = 'Helper' Or
account_type = 'Resident/Employee')"

    End If

    ' Load From DB
    GlobalFunctions.db_connect()

    Dim reader As MySqlDataReader

    Dim command As MySqlCommand = connection.CreateCommand()
    command.CommandText = sql
    reader = command.ExecuteReader()
    If reader.HasRows Then

        Do While (reader.Read())
            'send sms
            sms = "Someone is attempting to enter the Building Premises"

```

```

        strnumber = (reader("contact").ToString) 'recipeint
        Call frmMain.SEND_SMS(strnumber, sms)
    Loop

    ' MsgBox("Messages sent!!")
    GlobalFunctions.connection.Close()

End If

End Sub

Public Sub OPEN_DOOR()
    'Door 1
    If cboDoor.SelectedIndex = 0 Then

        If account_type = "Admin" Or account_type = "HR" Or account_type =
"Owner" Or account_type = "Employee" Or account_type = "Resident" Or account_type
= "Guard" Or account_type = "Helper" Or account_type = "Resident/Employee" Then
            Call LOG_ACCOUNT() 'look up if login /log out

            Out(&H378, 8)
            Application.DoEvents()
            System.Threading.Thread.Sleep(300)
            Out(&H378, 0)
            ' MsgBox("Open Door 1")
        Else
            lblStatus.Text = "Your not allowed to Enter to this door!" ' set
status
            ATTEMPT_LOG()

            'login 3 time fail then send sms
            door1_fail_log = door1_fail_log + 1
            If door1_fail_log = 3 Then
                door1_fail_log = 0 'sequence return to zero
                ' MsgBox("You have used 3 attempt!this window will be
closed!", vbCritical, "-Error")
                Call SEND_NOTIFICATIONS()

            End If
            door1_fail_log = door1_fail_log

        End If

    End If

    'Door 2
    If cboDoor.SelectedIndex = 1 Then

        If account_type = "Admin" Or account_type = "HR" Or account_type =
"Owner" Or account_type = "Employee" Or account_type = "Resident/Employee" Then
            Call LOG_ACCOUNT() 'look up if login /log out
            Out(&H378, 2)
            Application.DoEvents()

```

```

        System.Threading.Thread.Sleep(300)
        Out(&H378, 0) ' output decimal to printerport 0000 0000
        ' MsgBox("Open Door 2")
    Else
        lblStatus.Text = "Your not allowed to Enter to this door!" ' set
status
        ATTEMPT_LOG()

        'login 3 time fail then send sms
        door2_fail_log = door2_fail_log + 1
        If door2_fail_log = 3 Then
            door2_fail_log = 0 'sequence return to zero
            ' MsgBox("You have used 3 attempt!this window will be
closed!", vbCritical, "-Error")
            Call SEND_NOTIFICATIONS()

            End If
            door2_fail_log = door2_fail_log

        End If

    End If

    'Door 3
    If cboDoor.SelectedIndex = 2 Then

        If account_type = "Owner" Or account_type = "Resident" Or account_type
= "Helper" Or account_type = "Resident/Employee" Then
            Call LOG_ACCOUNT() 'look up if login /log out
            Out(&H378, 16)
            Application.DoEvents()
            System.Threading.Thread.Sleep(300)
            Out(&H378, 0) ' output decimal to printerport 0000 0000
            ' MsgBox("Open Door 3")
        Else
            lblStatus.Text = "Your not allowed to Enter to this door!" ' set
status
            ATTEMPT_LOG()

            'login 3 time fail then send sms
            door3_fail_log = door3_fail_log + 1
            If door3_fail_log = 3 Then
                door3_fail_log = 0 'sequence return to zero
                ' MsgBox("You have used 3 attempt!this window will be
closed!", vbCritical, "-Error")
                Call SEND_NOTIFICATIONS()

                End If
                door3_fail_log = door3_fail_log

            End If

        End If

    End If

End Sub

```

```

Public Sub ATTEMPT_LOG()
    ' Connect to Database
    GlobalFunctions.db_connect()
    Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()
    Try
        command = New MySqlCommand("INSERT INTO attempt_logs values('','" &
txtID.Text & "',''" & txtname.Text & "',''" & Me.txtAccountType.Text & "',''" &
Date.Today & "',''" & TimeOfDay & "',''" & Me.cboDoor.Text & "') ",
GlobalFunctions.connection, transaction)
        command.ExecuteNonQuery()
        transaction.Commit()

        Catch ex As MySqlException
            MessageBox.Show("Error in inserting new data! Error: " & ex.Message,
>Data Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            transaction.Rollback()
        End Try
        'close connections
        GlobalFunctions.connection.Close()

    End Sub

    Private Sub frmDTR_Load_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

```

```

Imports System.Windows.Forms
Imports MySql.Data.MySqlClient
Public Class frmLogs

```

```

    Private Sub frmLogs_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.DateTimePicker1.Value = Today.Date

```

```

    End Sub

```

```

Public Sub LOAD_LOGS()

    ' Connect to Database
    GlobalFunctions.db_connect()
    Try
        Dim Adapter As New MySqlDataAdapter("SELECT * FROM account_logs WHERE
Date='" & Me.DateTimePicker1.Value.Date & "' ORDER BY detail_id DESC",
GlobalFunctions.MySQLConnectionString)
        Dim dt As New DataTable("accounts")
        Adapter.Fill(dt)
        Me.DataGridView1.DataSource = dt

        With Me.DataGridView1

```

```

        ' .Columns(0).HeaderText = "Entry ID"
        ' .Columns(1).HeaderText = "ID Number"
        ' .Columns(2).HeaderText = "Full Name"
        ' .Columns(3).HeaderText = "Date"
        ' .Columns(4).HeaderText = "Time IN"
        ' .Columns(5).HeaderText = "Time OUT"

    End With

Catch ex As MySqlException

    MessageBox.Show("Error " & ex.ToString, "Loading Records")

    'close connections
    GlobalFunctions.connection.Close()

End Try

End Sub
Public Sub LOAD_ATTEMPT_LOGS()

    ' Connect to Database
    GlobalFunctions.db_connect()
    Try
        Dim Adapter As New MySqlDataAdapter("SELECT * FROM attempt_logs WHERE
Date='" & Me.DateTimePicker1.Value.Date & "' ORDER BY detail_id DESC",
GlobalFunctions.MySQLConnectionString)
        Dim dt As New DataTable("accounts")
        Adapter.Fill(dt)
        Me.DataGridView1.DataSource = dt

        With Me.DataGridView1

            ' .Columns(0).HeaderText = "Entry ID"
            ' .Columns(1).HeaderText = "ID Number"
            ' .Columns(2).HeaderText = "Full Name"
            ' .Columns(3).HeaderText = "Date"
            ' .Columns(4).HeaderText = "Time IN"
            ' .Columns(5).HeaderText = "Time OUT"

        End With

    Catch ex As MySqlException

        MessageBox.Show("Error " & ex.ToString, "Loading Records")

        'close connections
        GlobalFunctions.connection.Close()

    End Try

End Sub

Private Sub DateTimePicker1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs)
    MessageBox.Show("Date Selected: " & DateTimePicker1.Value.Date)

```



```

End Sub

Public Sub DELETE_LOGS()

    If Me.CboLogType.SelectedIndex = 0 Then
        sql = "Truncate account_logs"
    End If

    If Me.CboLogType.SelectedIndex = 1 Then
        sql = "Truncate attempt_logs"
    End If

    ' Connect to Database
    GlobalFunctions.db_connect()
    Dim command As MySqlCommand
    Dim transaction As MySqlTransaction
    transaction = GlobalFunctions.connection.BeginTransaction()
    Try

        command = New MySqlCommand(sql, GlobalFunctions.connection,
transaction)
        command.ExecuteNonQuery()
        transaction.Commit()

        MsgBox("Logs Deleted!!!")

    Catch ex As MySqlException
        MessageBox.Show("Error in DELETING data! Error: " & ex.Message, "Data
Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        transaction.Rollback()
    End Try
    'close connections
    GlobalFunctions.connection.Close()

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim ans As String

    ans = MsgBox("Would you like to Clear all Logs?", vbYesNo)
    If ans = vbYes Then

        Call DELETE_LOGS()

    Else

    End If
End Sub

```

```

        Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
            Me.Close()
        End Sub

        Private Sub DateTimePicker1_ValueChanged_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DateTimePicker1.ValueChanged
            If Me.CboLogType.SelectedIndex = 0 Then
                LOAD_LOGS()
            End If

            If Me.CboLogType.SelectedIndex = 1 Then
                LOAD_ATTEMPT_LOGS()
            End If
        End Sub

        Private Sub CboLogType_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CboLogType.SelectedIndexChanged
            If Me.CboLogType.SelectedIndex = 0 Then
                LOAD_LOGS()
            End If

            If Me.CboLogType.SelectedIndex = 1 Then
                LOAD_ATTEMPT_LOGS()
            End If
        End Sub

        Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label2.Click
        End Sub
    End Class

Imports MySql.Data.MySqlClient
Public Class frmIndDTR

    Private Sub frmIndDTR_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Call LOAD_INDV_LOGS__REPORT()
    End Sub
    Public Sub LOAD_INDV_LOGS__REPORT()

        ' Connect to Database
        GlobalFunctions.db_connect()
        Try
            sql = "SELECT * FROM account_logs WHERE Id_number='" & Me.txtID.Text &
"" ORDER BY detail_id DESC"

```

```

        Dim Adapter As New MySqlDataAdapter(sql,
GlobalFunctions.MySQLConnectionString)
        Dim dt As New DataTable("accounts")
        Adapter.Fill(dt)
        Me.DataGridView1.DataSource = dt

Catch ex As MySqlException

        MessageBox.Show("Error " & ex.ToString, "Loading Records")

        'close connections
        GlobalFunctions.connection.Close()

End Try

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Me.Close()

End Sub

Private Sub cmdprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdprint.Click
    DataGridViewPrinter.StartPrint(DataGridView1, True, True, "Individual DTR
Report Generated", "Mapua")
End Sub

Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click

End Sub
End Class

Imports Microsoft.VisualBasic
Imports MySql.Data.MySqlClient
Public Class frmChange

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
        Me.Close()

End Sub
'check duplicated values
Sub if_EXIST()
    ' Load From DB
    GlobalFunctions.db_connect()

    Dim reader As MySqlDataReader

    Dim command As MySqlCommand = connection.CreateCommand()
    command.CommandText = "SELECT * from account_records where password='" &
Me.txtOldpAss.Text & "' AND account_type='" & account_type & "'"

```

```

        reader = command.ExecuteReader()
        If (reader.HasRows) Then
            While (reader.Read())

                End While
                UPDATE_ADMIN()

            Else
                GlobalFunctions.connection.Close()
                MsgBox("Wrong Credentials!!", MsgBoxStyle.Critical)
                Me.txtOldpAss.Text = ""
                Me.txtOldpAss.Focus()

                ClearTextBox(Me)
            End If
        End Sub
        Sub UPDATE_ADMIN()

            If Me.TXTnewpass.Text = txtRetype.Text Then

                ' Connect to Database
                GlobalFunctions.db_connect()
                ' Dim command As MySqlCommand
                Dim transaction As MySqlTransaction
                transaction = GlobalFunctions.connection.BeginTransaction()

                Try
                    GlobalFunctions.execute_nonquery("Update account_records set
password='" & Me.TXTnewpass.Text & "' WHERE password='" & Me.txtOldpAss.Text & "'
AND account_type='" & account_type & "' ")
                    transaction.Commit()

                    MsgBox("Account Successfully Updated!!", MsgBoxStyle.Information)

                    ClearTextBox(Me)
                    Me.Close()
                    Me.Dispose()

                Catch ex As MySqlException
                    MsgBox.Show("Error in Updating data! Error: " & ex.Message,
"", MessageBoxButtons.OK, MessageBoxIcon.Error)
                    transaction.Rollback()
                End Try
                ' close connections
                GlobalFunctions.connection.Close()
            Else
                MsgBox("Retype password does not match with your New Psssword!!")
                Me.txtRetype.Text = ""
                Me.txtRetype.Focus()

            End If

        End Sub
        Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click

```

```

        If Me.txtOldpass.Text = "" Or Me.TXTnewpass.Text = "" Or Me.txtRetype.Text
= "" Then
            MsgBox("Incomplete Detail!!", MsgBoxStyle.Critical)
            Exit Sub

        End If
        if_EXIST()
    End Sub

    Private Sub frmChange_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        End Sub

    Private Sub Label15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label15.Click

        End Sub
    End Class

Imports MySql.Data.MySqlClient
Public Class frmReports

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Me.Close()

    End Sub

    Private Sub cmdprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdprint.Click
        DataGridViewPrinter.StartPrint(DataGridView1, True, True, "Report
Generated", "Mapua")
    End Sub

    Private Sub CboAcctType_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CboAcctType.SelectedIndexChanged
        Call LOAD_LOGS__REPORT()
    End Sub

    Private Sub frmReports_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        If account_type = "Owner" Then
            With CboAcctType
                .Items.Clear()
                .Items.Add("All")
                .Items.Add("Admin")
                .Items.Add("Owner")
                .Items.Add("HR")
                .Items.Add("Employee")
                .Items.Add("Resident")
                .Items.Add("Guard")
                .Items.Add("Helper")
                .Items.Add("Resident/Employee")
            End With
        End If
    End Sub

```

```

If account_type = "HR" Then

    With CboAcctType
        .Items.Clear()
        .Items.Add("Employee")
    End With

End If

If account_type = "Admin" Then

    With CboAcctType
        .Items.Clear()
        .Items.Add("All")
        .Items.Add("Admin")
        .Items.Add("Owner")
        .Items.Add("HR")
        .Items.Add("Employee")
        .Items.Add("Resident")
        .Items.Add("Guard")
        .Items.Add("Helper")
        .Items.Add("Resident/Employee")

    End With

End If

End Sub
Public Sub LOAD_LOGS__REPORT()

    ' Connect to Database
    GlobalFunctions.db_connect()
    Try
        sql = "SELECT * FROM account_logs WHERE Account_type='" &
Me.CboAcctType.Text & "' ORDER BY detail_id DESC"

        If Me.CboAcctType.Text = "All" Then
            sql = "SELECT * FROM account_logs ORDER BY Account_type DESC"
        End If

        Dim Adapter As New MySqlDataAdapter(sql,
GlobalFunctions.MySQLConnectionString)
        Dim dt As New DataTable("accounts")
        Adapter.Fill(dt)
        Me.DataGridView1.DataSource = dt

    Catch ex As MySqlException

        MessageBox.Show("Error " & ex.ToString, "Loading Records")

        'close connections
        GlobalFunctions.connection.Close()
    End Try
End Sub

```

```

        End Try

    End Sub

    Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click

        End Sub
    End Class

Public Class EnhancedPrintPreviewDialog
    Inherits System.Windows.Forms.PrintPreviewDialog

    Dim ShowCounter As Boolean = True

    Private Sub myPrintPreview_Shown(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Shown

        If ShowCounter = True Then

            'Get the toolstrip from the base control
            Dim ts As ToolStrip = CType(Me.Controls(1), ToolStrip)

            'Get the print button from the toolstrip
            Dim printItem As ToolStripItem = ts.Items("printToolStripButton")

            'Add a new button (our own) to the toolstrip by copying properties
from the original
            With printItem

                Dim myPrintItem As ToolStripItem
                myPrintItem = ts.Items.Add(.Text, .Image, New
EventHandler(AddressOf MyPrintItemClicked))

                'Alter as many other personalized stuff here
                myPrintItem.DisplayStyle = ToolStripItemDisplayStyle.Image

                'Relocate the item to the beginning of the toolstrip
                ts.Items.Insert(0, myPrintItem)

            End With

            'Remove the original button
            ts.Items.Remove(printItem)

            ShowCounter = False

        End If

    End Sub

    Private Sub MyPrintItemClicked(ByVal sender As Object, ByVal e As EventArgs)

        'Allow the user to choose a printer and specify other settings.
        Dim dlgPrint As New PrintDialog

```

```

        With dlgPrint
            .Document = Me.Document
            '.UseEXDialog = True
            .AllowSelection = False
            .ShowNetwork = False
            .AllowCurrentPage = True
            .AllowSomePages = True
        End With

        'If the user clicked OK, print the document.
        If dlgPrint.ShowDialog = Windows.Forms.DialogResult.OK Then
            Me.Document.Print()
        End If

    End Sub

    ' Private Sub InitializeComponent()
    '     Me.SuspendLayout()
    '
    '     'EnhancedPrintPreviewDialog
    '
    '     Me.ClientSize = New System.Drawing.Size(806, 384)
    '     Me.Name = "EnhancedPrintPreviewDialog"
    '     Me.ResumeLayout(False)

    '     End Sub

    Private Sub EnhancedPrintPreviewDialog_Load(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Function Document() As Printing.PrintDocument
        Throw New NotImplementedException
    End Function

End Class

Public Class frmHelp

    Private Sub cmdClose_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles cmdClose.Click
        Me.Close()

    End Sub

    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As
        System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
        TextBox1.Text = "Indoor Biometric Security with SMS Alert System"
    End Sub

    Private Sub LinkLabel3_LinkClicked(ByVal sender As System.Object, ByVal e As
        System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel3.LinkClicked

```



```

        TextBox1.Text = "Student Number: 2007180004" & vbCrLf & "Address: 1245A
Craig Street Sampaloc Manila" & vbCrLf & "Birthday: October 21, 1989"
    End Sub

    Private Sub LinkLabel4_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel4.LinkClicked
        TextBox1.Text = "Student Number: 2006" & vbCrLf & "Address: " & vbCrLf &
"Birthday:"
    End Sub

    Private Sub LinkLabel2_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel2.LinkClicked
        TextBox1.Text = "Student Number: 2005" & vbCrLf & "Address: " & vbCrLf &
"Birthday:"
    End Sub

    Private Sub LinkLabel5_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel5.LinkClicked
        TextBox1.Text = "COE Professor" & vbCrLf & "-----" & vbCrLf & "-----"
    End Sub

    Private Sub LinkLabel6_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel6.LinkClicked
        TextBox1.Text = "Design Professor" & vbCrLf & "-----" & vbCrLf & "-----"
    End Sub

    Private Sub Label6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label6.Click

    End Sub
End Class

```

Indoor Biometric Security with SMS Alert System and Electronic Logbook

John Michael Bernabe, Jessica Mae Salgado, Mirriam Joy Sorreda

Ayra Panganiban, Adviser

BS in Computer Engineering – School of EECE, Mapua Institute of Technology

Abstract— The main goal of the group is to design a security system that addresses to the drawbacks of using the traditional system of security. In this documentation, the researchers present an alternative method of security by using fingerprint scanners. The system consists of a biometric fingerprint scanner, electric door lock, software, and SMS module. The software will be responsible to operate the scanner and the door lock. The program is installed in a desktop computer running on a Windows operating system. A broadband is used for the SMS notification of the system. The design is capable of having levels of access which will restrict users to enter places when they don't have the authorization. Log ins and log outs will be recorded into the database for reporting purposes.

Keywords— Biometric Fingerprint Scanner, Electronic Door Lock, Software, SMS Module, Windows Operating System,

I. INTRODUCTION

Safety is the condition or state of having the freedom from failure, damage, error, accidents, harm, injury, loss or any event considered to be undesirable. In order to be safe, one has to be secured. Security is the form of protection against any undesirable events. In order to prevent any risks of danger, harm, or loss one must have security. Some examples or forms of securities include door locks, security guards, anti-virus applications, and passwords.

In terms of security, nowadays, biometric authentication is considered as more reliable compared to the traditional security such as password based and lock-and-key. Biometric comes from the Greek words “bios” and “metron” which means life and measurement. It refers to the technology of identifying and/or authenticating a person using distinct human body characteristics such as face, fingerprint, DNA, palm print, iris, retina, and voice. This biological identification technology provide higher sense of security compared to the traditional ones for the reason that these unique marks or features in the body cannot be given to someone else to use and these cannot be lost or misplaced for it is something that people have all the time. To consider a biometric trait as a reliable and secured option for authentication, it should possess universality, distinctiveness, permanence, and collectability. Other criteria for assessment are performance, acceptability, and circumvention.

Among all the biometric traits, fingerprint is one of the oldest methods used for various practices. Dating back in the 14th century, China used fingerprints to distinguish one individual to another. Fingerprint identification is the most widely used of all the biometric devices because of its uniqueness and consistency over time. Applications of biometrics include computer login, access to office buildings and homes, protecting personal property and etc.

II. REVIEW OF RELATED DESIGN LITERATURE AND STUDIES

From the article entitled “A Guide to Biometric Fingerprint Sensors: Major Manufacturers and Technical Specifications” by Michelle Shen, an IT Consultant of ePolyMath.com, the researchers were able to learn the different types of fingerprint sensors and their costs. From that, we learned that among all the sensors, semiconductor sensors are considered to be low cost, optical sensors are considered to have a high degree of stability and reliability, while ultrasound sensors are very precise and fraud-free though expensive to implement.

Also, from Michelle Shen’s “Vendor Fingerprint Sensors Comparison Chart”, the details of the technical specifications of the sensor were discussed as well as those so-called fingerprint application modules which contain fingerprint sensor, middleware and the like. The information in the chart gave the researches a handy guide to better understand what other developers have achieved, what they were doing and where they are moving to.

Another article related to the design which was the “Fingerprint sensor with feature authentication” written by David Kinsella presents information on how a fingerprint sensor works. Through this article, researchers were able to acquire knowledge about how the device reads the fingerprint of a person. Researchers were able to see how detection, analysis and authentication work for the fingerprint sensor.

III. DESIGN PROCEDURES

This chapter gives a detailed procedure for developing the software and hardware of the prototype.

A. Hardware Development

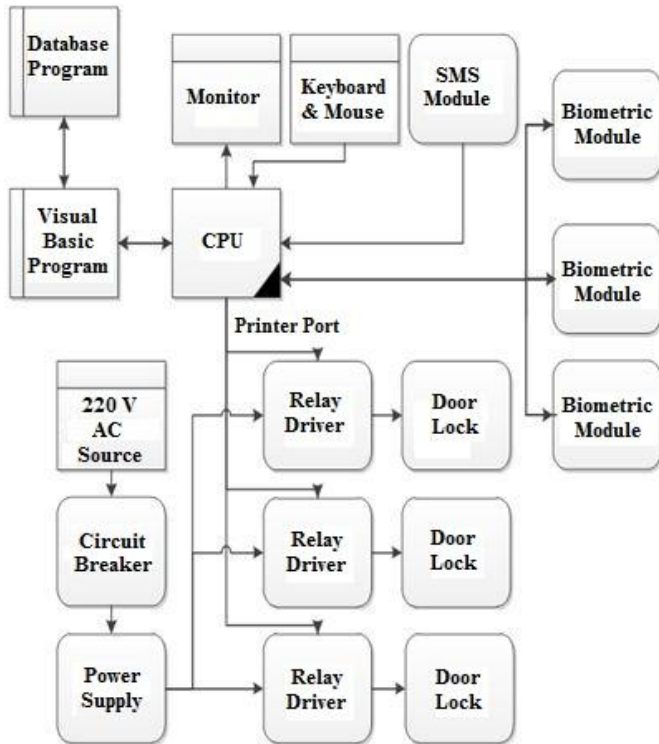


Fig. 1 Block Diagram

The main system worked with Visual basic application, it also had a database linked using SQL. The system functioned with a computer desktop wherein the three biometrics were connected. An SMS module was also installed in the computer desktop for notification purposes. With the use of a parallel data cable, three relay drivers were connected, also supplied with a 220V power source. The relay drivers were then connected to the electronic door lock to supply the required amount of current needed to trigger the door lock to open. A circuit breaker was used to protect electrical circuit from damage due to short circuit or overloading.

B. Software Development

For the programming part, the group will be using the language Visual Basic (software) and will link it to SQL (software) for the database. An administrator account will be created for the configuration of the registration of new users. The program can add, edit, update, delete, and remove all users from the system. The administrator can also modify the accounts

in the system, change passwords, monitor the logs of all the tenants in the building and generate reports. There will also be an account for human resource purposes such as the attendance monitoring of the employees at the office for evaluation and printed reports. In addition to that, the design will be having an account solely for the owner that will have the access to view all the records and can print all date and time records. A user can only have 3 attempts to log-in using the biometric scanner. If limit is reached, there will be an SMS sent to the owner or to the administrator or to a recipient who wished to receive notifications to inform that there is someone who is attempting to enter the building or different floor levels. This attempt will also be logged in the database. A personal computer will be installed at the office located at the second floor of the building.

As regards the levels of access, each person type will be given a specific kind of authorization. All person types, administrator, owner, Human Resource manager, guard, employee, resident, and helper will only be able to access specific areas. The sole person type who has the access on all floors is the owner. While the Human Resource manager and employee can only go through the main gate and the office door. The resident and helper person types will only have access to the main gate and residential door.

IV. TESTING, PRESENTATION AND INTERPRETATION OF DATA

This chapter shows the different tests done by the group in line with the objectives enumerated in the first chapter. The testing was done after the development and construction of the prototype to verify if the objectives were accomplished.

A. Failure to Enroll Rate

The first thing needed for the design to serve its purpose is to store data assigned to the user in which area he / she is allowed to enter. The test is done using the registration of the user by assigning his/her account in different account type to the registration form.

TABLE I
FAILURE TO ENROLL RATE

User	Trial 1	Trial 2	Trial 3
Kim	Success	Success	Success
Ralph	Success	Success	Success
Ian	Success	Success	Success

Sedney	Success	Success	Success
Ken	Success	Success	Success
Hazel	Success	Success	Success
Janica	Success	Success	Success
Kevin	Success	Success	Success
Lorise	Success	Success	Success
Nico	Success	Success	Success

The table above shows the registration of a user using the digital persona biometric where the system has high enrollment rate. All registration are successful after placing the users' finger in the scanner four times. The registration on the fingerprint will repeat its process, if the user registration is not successful. This will just finish its process once the registration is successful.

B. Reading Test

Once there is/are successful registrants, when they tap their finger/s already registered, the program will now read the fingerprint and will welcome the user/s. This compares a person's fingerprint to another one previously recorded in the system database. The computer locates the previous fingerprint by comparing two fingerprints recorded. Once matched from the one recorded previously, the door lock will be activated and opened.

TABLE III
FAILURE TO CAPTURE RATE

USERS	Kim	Ralph	Ian	Sedney	Ken	Hazel	Janica	Kevin	Lorise	Nico
Kim	S	F	F	F	F	F	F	F	F	F
Ralph	F	S	F	F	F	F	F	F	F	F
Ian	F	F	S	F	F	F	F	F	F	F
Sedney	F	F	F	S	F	F	F	F	F	F
Ken	F	F	F	F	S	F	F	F	F	F
Hazel	F	F	F	F	F	S	F	F	F	F
Janica	F	F	F	F	F	F	S	F	F	F
Kevin	F	F	F	F	F	F	F	S	F	F
Lorise	F	F	F	F	F	F	F	F	S	F
Nico	F	F	F	F	F	F	F	F	F	S

Table II shows the results of the test done using the failure to capture the rate. This performance metric is the probability the system fails to detect a biometric input when presented correctly. The result clearly shows reading a fingerprint using the digital persona biometric has high capture rate of the design. This shows biometric scanner has a 100% capture rate in determining whether a fingerprint is enrolled in a system or not.

TABLE IIIII
FAILURE TO CAPTURE RATE WITH CONDITIONS

Finger Condition Accuracy Test					
USERS	Correct Alignment of Finger	Incorrect Alignment of Finger	Stained or Dirt on the Finger	Foreign Particle Present on the Finger	Wet Finger
Kim	Success	Success	Success	Success	Fail
Ralph	Success	Success	Success	Success	Fail
Ian	Success	Success	Success	Fail	Fail
Sedney	Success	Success	Success	Success	Fail
Ken	Success	Success	Success	Success	Fail
Hazel	Success	Success	Success	Fail	Fail
Janica	Success	Success	Success	Success	Fail
Kevin	Success	Success	Success	Success	Fail
Lorise	Success	Success	Success	Success	Fail
Nico	Success	Success	Success	Fail	Fail

From Table III, it shows scanning of a correct alignment of finger has high rate acceptance using the fingerprint scanner of the design. All scanning of fingerprints are successful in all trials from different users.

The second test is the incorrect alignment of finger to be scanned by the fingerprint scanner. Some biometric scanners cannot recognize the user's fingerprint with an incorrect alignment of finger to be scanned. With this test, it shows that the digital persona biometric scanner has a high capture rate that can read and can identify the registered user even if the users place their finger in an incorrect alignment.

On the third test, it shows that the stained/dirt on the finger has a high rate using the fingerprint scanner. This stained/dirt on the thumb is not the problem to have a successful log in or log out using the biometric scanner because there is always a chance that user's finger is dirty or stained in some situations.

The fourth test shows the test with any foreign particle on the thumb using the fingerprint scanner has a good capture rate. In this test the biometric can also read and identify if the user is registered or not, even if there is a foreign particle in the user's thumb or scanner. To avoid this problem make sure the biometric scanner is clean before scanning the finger.

The last acceptance test shows wet finger causes the fingerprint scanner's accuracy very low or bad as the finger is impossible to read. To avoid this kind of problem users must remember the finger must be dried to have a successful log in / log out to the scanner.

The results show all the tests from the fingerprint scanner will also vary from the fingerprint conditions. It also shows all tests except wet finger will have a high capture rate of the fingerprint scanner using the digital persona biometric scanner.

TABLE IV
LOG IN/LOG OUT ACCURACY TEST

Log In /Log Out Test									
Finger Print #	Trial 1			Trial 2			Trial 3		
	Scan	Door	SMS	Scan	Door	SMS	Scan	Door	SMS
1	Success	Open	None	Success	Open	None	Success	Open	None
2	Success	Open	None	Success	Open	None	Success	Open	None
3	Success	Open	None	Success	Open	None	Success	Open	None
4	Fail	Close	Sent	Success	Open	None	Success	Open	None
5	Success	Open	None	Success	Open	None	Success	Open	None
6	Fail	Close	Sent	Success	Open	None	Success	Open	None
7	Success	Open	None	Success	Open	None	Success	Open	None
8	Fail	Close	Sent	Success	Open	None	Success	Open	None
9	Success	Open	None	Success	Open	None	Success	Open	None
10	Success	Open	None	Success	Open	None	Success	Open	None

The test shows in Table IV how the door lock, SMS module, and fingerprint scanner interact with each other. The fingerprint is responsible to send signals or data from the program, triggers the door lock, or sends SMS alert to the respondent. This table also shows how the door lock and SMS alert response in the result of the fingerprint scanner to the scanning of the thumb of the user.

V. CONCLUSION AND RECOMMENDATION

A. Conclusion

In the design Indoor Fingerprint Security with SMS alert and Electronic Logbook, the designers were able to build a system to address the needs of the customer, Mr. Jhayson Allen S. Tan. The group was able to design a system that would give a better and tighter security to the occupants in the compound. The system built is a PC interfaced fingerprint scanner with a desktop application and SMS module. With this system, only authorized people will be able to pass through the main gate and specific areas of the compound. Levels of access were imposed in the system where the users would only be able to enter the areas assigned to them. Through these levels of access implemented on the biometric sensor, it would help control the access of people coming in and out of the building. Using the electronic logbook, this would register and record the log in and log out information of the person coming in and out of the building. People who were not registered in the database of the program could not enter the building premises.

Moreover, an SMS alert system was incorporated in the system to notify the chosen respondents if there was/were unauthorized person/s who attempted to enter a specific area of the building.

B. Recommendation

To attend to the delimitations of the project prototype, the researchers recommend the enhancement of the project by making the transmission of data wireless so that it is less prone to tampering and it will definitely eliminate the use of long wires and cables. The project supports only fingerprint scanning, so the enhancement may well be as adding a new feature such as eye scanner. Use a USB converter instead of serial data cables and DB-25 pin data cable since most of the CPU's nowadays have few serial ports and sometimes, no printer ports. For the SMS alert system, the researchers recommend the use of gateway instead of modem. However, an internet connection is needed in using the gateway. Compared to the SMS alert system using the modem, the sending of message through a gateway will be faster and will not affect the performance of the computer.

VI. REFERENCES

- Adhami, R., Meenen, P (2001). Fingerprinting for Security. *Potentials*, 20, pg. 33-38
- Dass, S.C., Jain A.K., Yongfang Zhu (2006). Validating a Biometric Authentication System: Sample Size Requirements. *Pattern Analysis and Machine Intelligence*, 28, pg. 1902-1319
- Faundez-Zanuy, Marcos (2004). A Door-Opening System Using A Low-Cost Fingerprint Scanner and a PC. *IEEE A&E SYSTEMS MAGAZINE*.
- Maltoni, D., Maio, D., Jain, A. K., Prabhakar S. (2009). *Handbook of Fingerprint Recognition*, Second Edition, Springer.
- Wertheim, K.E. (2010). Human Factors in Large-Scale Biometric Systems: A Study of the Human Factors Related to Errors in Semiautomatic Fingerprint Biometrics. *Systems Journal*, 4, pg. 138-146
- Herzog P. (2008, July 17). Home Security Methodology. Retrieved from <http://www.isecom.org/research/hsm.html>
- Hudson K. (2011, May 30). Biometric Embedded Fingerprint Reader Modules. Retrieved from <http://www.articlesbase.com/security-articles/biometric-embedded-fingerprint-reader-modules-4834241.html>
- Shen M. (2002, September 16). A Guide to Biometric Fingerprint Sensors: Major Manufacturers and Technical Specifications. Retrieved from <http://www.tmcnet.com/biomag/features/shen0902.htm>